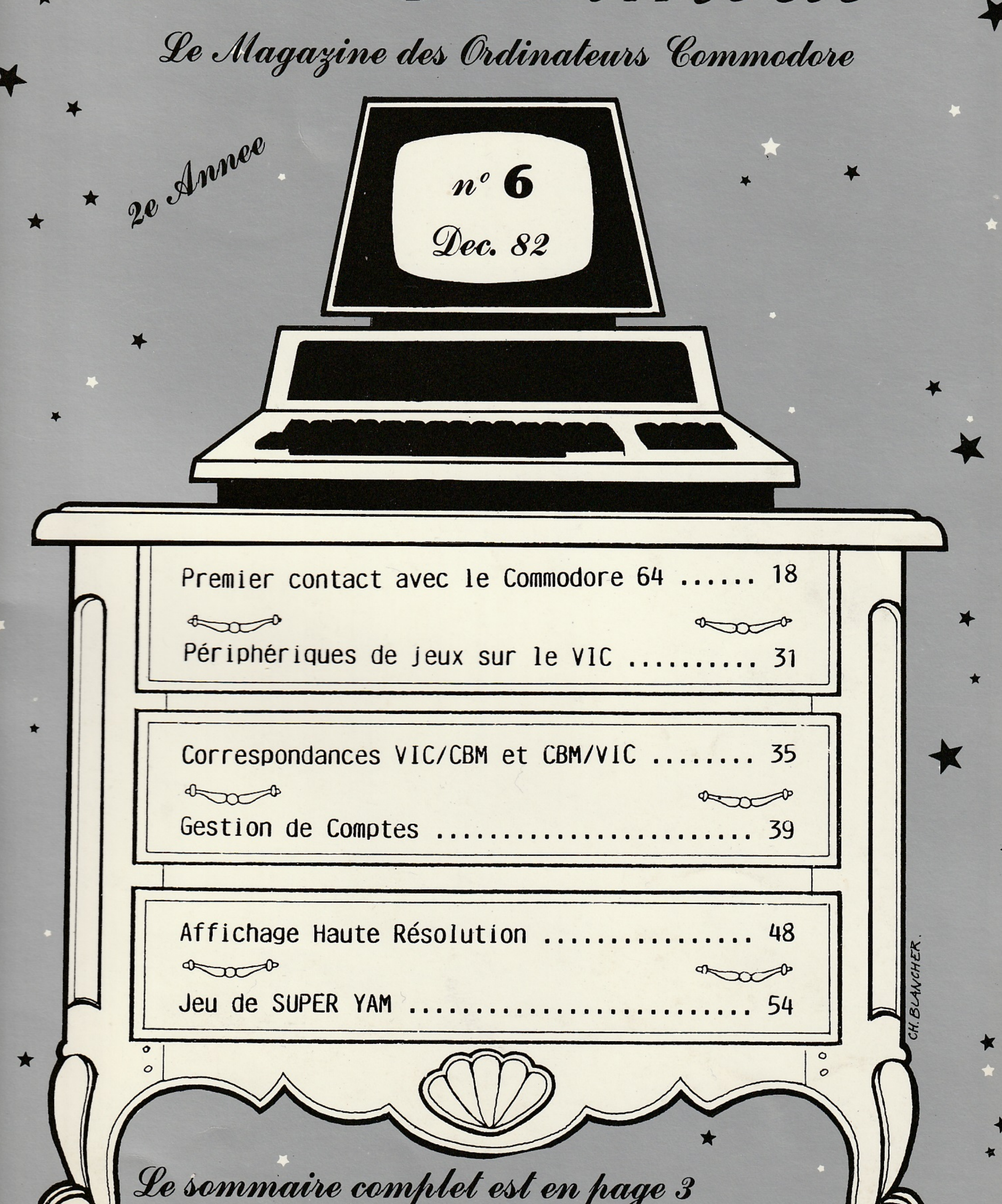
La Commode

Le Magazine des Ordinateurs Commodore

2e Annee



Le sommaire complet est en page 3

35 Francs



VIC 20 : L'ORDINATEUR-COPAIN.

Avec le Commodore Vic 20 vous vous faites 4 copains d'un seul coup.

Un copain pour le jeu; Vic 20 est champion : Échecs, Guerre des Étoiles, Alien, Les Envahisseurs..., vous passerez des heures avec lui.

Un copain pour les études; Vic 20 vous apprend les maths ou les statistiques ou l'anglais..., en vous amusant.

Un copain pour l'initiation à l'informatique; pas de problème, un cours pratique d'autoformation vous apprendra le langage informatique. Vous pourrez créer ainsi vos propres programmes.

Un copain pour la gestion familiale; avec ses programmes de comptabilité, de gestion de fichiers et calculs électroniques, Vic 20 aidera toute la famille.

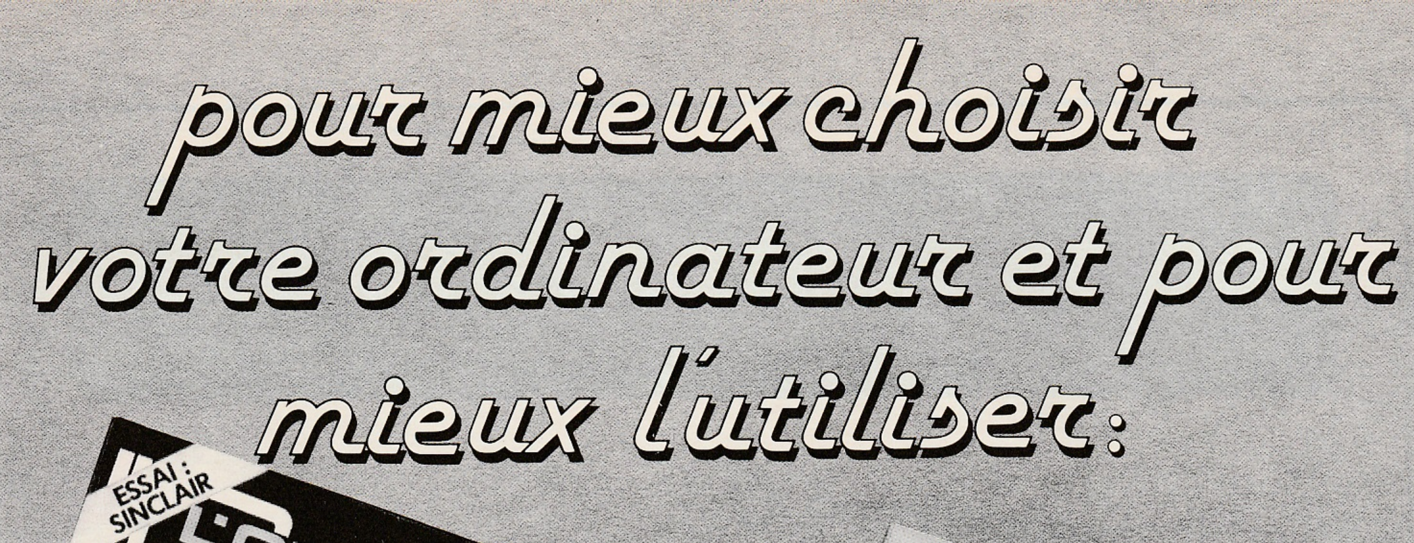
Il y a plus de 200 revendeurs Commodore en France. Venez les voir. Pour Noël, ils vous réservent une bonne surprise à un prix tout à fait amical.

COMMODORE VIC 20
LE COPAIN DE TOUTE LA FAMILLE.

Pour 3200 F* TTC seulement:
— Le Vic 20 avec interface noir et blanc intégré — Le lecteur de cassette — Le cours d'autoformation au Basic comprenant un manuel d'utilisation et 2 cassettes de travaux pratiques.
— Une cassette comprenant deux jeux : Blitz et Night Park.
— Une cassette comprenant deux programmes :
• 1 programme éducatif : Calcul Spatial ;
• 1 programme gestion : Budget Familial.

*Offre spéciale valable jusqu'au 31/12/82.

Sommaire



lisez

L'ORDINATEUR INDIVIDUEL

Vous y trouverez :

l'actualité et les tendances de l'informatique individuelle • les bancs d'essais des principaux matériels • des panoramas et des tests comparatifs • le point des grandes manifestations internationales • des articles d'initiation • des synthèses • des programmes • des interviews "exemplaires" • des conseils • des idées • des astuces

L'ORDINATEUR INDIVIDUEL, chez votre marchand de journaux

Abonnement pour l'Afrique (1 an, 11 numéros, envoi par avion) : 270 FF **Tarif Haute-Volta : 270 FF**

EDITORIAL

Nous avons commencé à dépouiller les feedback et ils se sont révélés très satisfaisants pour nous ; beaucoup d'articles ont eu une moyenne élevée, ce que confirment par ailleurs de nombreuses lettres du courrier des lecteurs.

Le feedback laisse toutefois apparaître une tendance un peu fâcheuse : il semble se dessiner un clivage entre les lecteurs possesseurs de VIC, qui ne lisent pas les articles destinés au CBM, et vice-versa. Il y a trop d'homogénéité dans la gamme Commodore pour que ceci se prolonge. Vous ne pouvez avoir que des articles destinés uniquement à votre machine précise. De plus, il est bien rare qu'il n'y ait pas à grappiller telle ou telle astuce dans un article, même s'il est destiné à une configuration autre que la vôtre.

En contrepartie de cet effort que -nous en sommes conscients- cela demandera de la part des Lecteurs, La Commode s'efforcera d'assurer qu'à la fin de chaque article figure un paragraphe ou deux expliquant comment adapter l'article à tous les modèles Commodore. Nous commençons avec l'article de Daniel TRECOURT sur la gestion de comptes. C'est dans cette même optique que la rubrique "Les Ficelles de La Commode" renfermera les astuces communes à toute la gamme.

Enfin, et la date de parution s'y prête, La Commode vous souhaite à tous de bonnes fêtes de Noël, et une bonne et heureuse anneée 1983.

Pierre-Etienne THALBERG

Les deux gagnants du feedback du numéro 1 sont Messieurs THERY et BASSABER. Ils recevront un stylo-montre.

Courrier des lecteurs

Nous possédons le système CBM 8001 :

- Unité centrale 8032 avec Edex 4,2
- Disquette 8050
- Imprimante 8024

Je me permets de vous poser les questions suivantes :

1) Pour compléter ou corriger un programme déjà assez long qui se trouve sur disquette, la procédure suivante est-elle la bonne :

a) charger le programme de la disquette en mémoire centrale (dload)

b) frapper au clavier les compléments ou les lignes corrigées du programme.

c) annuler le programme de la disquette (scratch)

d) réinscrire le nouveau programme sur la disquette (dsave)

2) N'y a-t-il pas moyen d'obtenir sur l'écran des traits continus avec plot x,y en verticale (en montant ou en descendant : on n'obtient que des traits discontinus.

3) Peut-on obtenir avec l'imprimante 8024 les lettres accentuées, les courbes réalisées sur l'écran avec plot ?

4) tab(x) sur l'imprimante semble avoir le même effet que spc(x) sur l'écran.

Y a-t-il moyen d'avoir pour l'imprimante une commande qui fournisse le même résultat que tab pour l'écran ?

5) Calcul de e à partir de

$$e = \lim_{m \to +\infty} (1 + 1/m)^m$$

Le calcul fonctionne correctement jusqu'à m=10.000 environ (qui donne e=2,71814671), puis donne des résultats aberrants pour les valeurs suivantes de m (ainsi pour m = 10.000.000, on obtient e = 2,74192959). Alors qu'une calculatrice TI-58 donne les résultats corrects jusqu'à la limite m=10 : e = 2,718281828).

Peut-on remédier à cette déficience du CBM 8032 ?
Par contre le calcul par :

$$e = \lim_{n \to +\infty} \sum_{i=0}^{n} 1/i!$$

marche correctement sur CBM
à n=13 on obtient e=2,71828183)

En vous remerciant,

Marcel GELLEY
29150 CHATEAULIN

1 - On peut condenser c) et d) en

faisant DSAVE "@NOM"

2 - On rend les lignes continues sur 8000 en faisant PRINT CHR$(142)

3 - Aucun graphique ou caractère programmable sur la 8024.

4 - Sur imprimante, TAB(x) fait x espaces. Il faut employer PRINT-USING de EDEX.

5 - Peut être lié à un problème d'erreur dans les exponentiations traité dans ce numéro ou le suivant, mais plus sûrement par le fait que dans la TI-58, les nombres sont représentés avec plus de chiffres. (les CBM n'en représentent que 9).

*

* *

Permettez-moi d'attirer votre attention sur la page 49 du dernier numéro de la Commode. Les adresses pour 4000 G ne sont pas très exactes, par exemple :

ACQ n'est point BE88 mais bien BD84
FLPINT n'est point C2EA mais bien C92D.

De même, des adresses en page zéro font que Basic + se plante à tous les coups. J'ai résolu ce petit problème en changeant les adresses des lignes numéros 26-27-28-29. J'ai remplacé les adresses en page zéro qui ne s'imposent pas par des adresses du buffer première cassette (je suis équipé de disquettes).

La partie PRINT @ ne veut rien savoir, je me suis acharné vainement dessus pendant quelques jours sans résultat. A bout d'idée, j'ai préféré la réécrire. J'y ai même gagné quelques octets en utilisant les routines BASIC, notamment la routine print.

Sorti de là, ce PGM est très bien et pour le Feed Back, je donnerai une note maximum à B. Michel (Pas complètement dénué d'arrières pensées, car j'attends avec impatience qu'il nous sorte un truc de la même eau !)

Puisque vous semblez désireux de savoir ce que vos lecteurs feront avec BASIC +, je puis vous dire que j'ai ajouté un titre (j'étais traumatisé au début par ce programme récalcitrant, aussi maintenant quand j'initialise ou réinitialise Basic +, l'écran se vide et un Rassurant BASIC + 4000 apparaît en haut de l'écran.

J'ai légèrement transformé le cadre pour qu'il prenne son motif dans une case mémoire (1ère cassette) ou qu'il utilise le motif original si l'on n' a rien mis dans la Case même.

J'ai aussi créé autre chose, si ça vous intéresse, ayez la bonté de me le signaler.

Par exemple :

SPEED REPETE (Repkey sauf 13)
OFF qui arrête Speed et REPETE
SOUND X,Y temps/60ème de seconde environ ! Il est à noter que Sound rend la main immédiatement et donc que le son continue pendant que le CBM fait autre chose.

SET viennnent de mon PGM
RESET (Commode numéro 5)
après transformation.

HEXA(X) X est un nombre décimal = affichage $YYYY
DECI(Y) est en préparation

Questions : 1) Pourriez-vous me dire pourquoi le CBM ne relit que 2 K d'une Eprom 2732 monotension. Avez-vous les brochages des 2532 et 2732 ?

2) Pourquoi dans le contexte de Basic +, la routine STROUT ne fonctionne qu'une fois et ensuite détruit le code source. Pouvez vous me dire quel paramètre important STROUT modifie-t-il ?

Bon, je vous remercie d'avance pour les réponses à mes questions que j'aimerai, déjà résolues !

Pouvez-vous dire à Monsieur Daniel Jean DAVID que lorsqu'on promet un bouquin (volume III : le système du PET/CBM) page 105 de la pratique du PET/CBM... et bien on tient ses promesses !

Remontrances faites en toute sympathie !...

Merci et meilleures salutations.

Jean-Paul LAURENT
13100 MARSEILLE

P.S. Quand publiez-vous une réalisation de stylo lumineux ???

Daniel Jean DAVID travaille sur le VIC, mais le livre paraîtra, c'est promis.

Les réalisations bricolées ne marchent pas très bien. Mais SIDEG en commercialise un tout fait.

*
* *

Permettez-moi de vous signaler une coquille, tant dans "la Pratique du PET/CBM" VOL.II page 116 que dans "LES CLEFS..." page 86, concernant l'adresse de la routine MULT dans le cas d'un CBM 4000 : il faut lire $ CB66 et non $CB61. Cela m'a causé quelques soucis, car $CB61 marche aussi, parfois, selon la valeur de Z. En effet, en $CB61 on trouve BNE CB66 suivi de JMP $CBC1, et, en CBC1, on a RTS... Peut être pourriez-vous signaler l'ERRATUM dans LA COMMODE.

PROGRAMME DE REFERENCES CROISEES EN BASIC

1 - Ce programme est bien pratique, même pour ceux qui possèdent un EDEX, car ce dernier ne permet pas de sortir la liste des variables utilisées dans un programme avec leur numéro de ligne, ni la liste des lignes appelées par GOTO, GOSUB, THEN ou RUN.

2 - Tel quel, il faut que le programme à analyser soit sur le drive 0. Il est facile de surmonter cette restriction en introduisant :

INPUT "DRIVE (0/1)" ; DR$

et en remplaçant le "0:" ligne 170 par DR$+":"

3 - J'avoue être intrigué par le code +",P,R" qui termine l'OPEN de cette même ligne 170. Je ne connaissais que +",S,R", ou rien du tout par défaut : c'est cette dernière solution que j'ai personnellement appliquée : mais +",P,R" marche aussi...

4 - Je suis également intrigué par la ligne 510 :

Que signifie : IF A$ =<> Q$ GOTO 490 ? Il semble que l'on aille à 490 quoiqu'il arrive : personnellement j'ai remplacé par :
510 GOTO 490 et n'ai vu aucune différence.

5 - Si, à la ligne 130, on choisit l'option "LIGNES", on obtient la liste des numéros de lignes appelées par GOTO GOSUB THEN et RUN, avec les numéros des lignes appelantes.

Mais les lignes appelées sont classées selon un mode décimal croissant, comme si elles étaient précédées de 0,... On voit par exemple se succéder dans l'ordre :

210
2153
300
30800
42

Je n'ai pas réussi à modifier le programme pour obtenir un classement dans l'ordre naturel.

6 - Il faut 3 mn 20 s pour obtenir la liste des variables du programme CROSSREF. Après compilation par l'intermédiaire du programme PET-SPEED, il ne faut plus que 20 s, soit 10 fois moins de temps. Dans le cas d'un gros programme (où l'intérêt de sortir la liste des variables est particulièrement grand). Il suffit par exemple de 12 minutes au lieu de 2 heures avec la version initiale. Je tiens cette version compilée à la disposition de ceux que celà intéresserait, pour échanger, ou, éventuellement, contre remboursement des frais de copie sur cassette ou disquette (type 8050).

J. PIERRAT
78000 VERSAILLES

3 - P signifie fichier programme.

Un F.P. est organisé exactement comme un séquentiel à part qu'il est marqué P ce qui fait que la routine LOAD l'accepte, et que les deux premiers octets forment l'adresse de chargement.
4 - c'est manifestement une erreur. Il faut lire :

510 IF A$ <> Q$ GOTO 490

En effet, Q$=CHR$(34) donc le code des guillemets, ce qui veut dire que si 510 GOTO 490 marche bien, vous ne mettez pas dans vos chaines de doubles guillemets.
5 - c'est tout à fait exact, le classement effectué n'est pas adapté aux numéros de lignes. Pour cela il faudrait classer le tableau X$ dans l'ordre croissant des B$ (voir ligne 580), après avoir transformé B$ en valeur numérique.

*
* *

Plutôt que de répondre au Feedback, je trouve préférable de vous indiquer ce que j'attends de votre Revue. Tel sujet qui ne vous intéresse pas un jour peut, très bien, vous intéresser demain ! Il serait donc dangereux, pour établir une statistique, de se baser sur une notation "instantanée".

Plus un débutant avancera dans la connaissance de la micro-informatique, plus il appréciera les articles qu'il ne pouvait intégrer au départ. Je vis, personnellement, cette démarche et je sais qu'il m'arrive de lire avec plaisir des articles que, naguère, je laissais de côté.

Je crois que la plupart de vos lecteurs sont des élèves qui attendent d'être instruits par les professeurs que vous êtes. Et, votre revue devrait être un complément aux livres sur les PET/CBM, rédigés par votre Directeur : Monsieur Daniel Jean DAVID -hélas, bien mal imprimés en caractères machine uniformes et pas toujours bien encrés!)-, suppléant aux insuffisances de PROCEP.

C'est pourquoi j'ai apprécié la page "Bibliographie", qui rectifie les erreurs contenues dans ces livres. Mais, j'apprécierais, aussi, des articles sur la manipulation des POKE et des PEEK, qui figurent souvent dans les progiciels dont j'ai fait l'acquisition ; ce serait un développement utile du Volume II de "La Pratique du PET/CBM...

Il existe, également, un problème lorsqu'on veut faire tourner un programme écrit pour 3000, sur un 4000, par exemple le progiciel PROCEP "Gestion de Fichier - Mailing - GF005/1". Votre article paru dans le numéro 2 de la COMMODE, page 36 ne donne pas assez de précisions pour passer de l'un à l'autre (n'y a-il pas un problème de pistes et de secteurs ?).

Mon équipement consiste en un 3032 avec 3040 et un Modem acoustique 8010 (modifié) relié à un IRIS 80, mais j'ai l'occasion de travailler, de temps en temps, sur un 4032 avec 4040. Arrivera-t-on, un jour ou l'autre, à une compatibilité entre marques, puisqu'il y a déjà des difficultés entre matériels proches d'une même marque ?

Etre autodidacte n'est pas très facile ! Je serais donc, très heureux de suivre des cours. Malheureusement, je ne bénéficie pas de la Formation Continue et je ne peux prendre à ma charge, au tarif prohibitif de ces cours, les frais d'inscription; vous pourriez, peut-être, connaitre un de ceux-ci, spécialisé sur COMMODORE, qui consent des prix "normaux"; auriez-vous, alors, l'amabilité de me communiquer son adresse.

Vous remerciant à l'avance, je vous prie de croire, Messieurs, à l'assurance de mes sentiments les meilleurs.

Pierre MONTAGNE
33500 LIBOURNE

1 Le seul problème de pistes et secteurs lorsqu'on passe de la série 3000 à la série 4000 est que ce ne sont pas tout à fait les mêmes nombres de secteurs qui sont disponibles. Dans La Pratique du PET/CBM vol. 1, on conseille de n'utiliser en accès direct que 15 ou 16 secteurs par piste et à ce moment, cela passe partout.

2 Le problème de séminaires bon

marché nous préoccupe. La Commode sera peut-être amenée dans l'avenir à organiser quelque chose dans ce sens.

*
* *

Je n'ai pas encore eu le temps d'étudier à fond tous les articles, (les quatre premiers numéros ne sont en ma possession que depuis une quinzaine de jours). Dès que cela sera fait, j'espère trouver un moment pour répondre au "feed-back" du numéro 4, et sans doute vous envoyer un ou deux programmes. Mais dès à présent, je tiens à vous exprimer toute ma satisfaction : LA COMMODE a bien répondu à mon attente.

Charles-Jean FERRERI
20228 LURI

Nous accepterons avec joie vos programmes, comme ceux de tous les autres lecteurs.

*
* *

Je possède un VIC 20 avec imprimante et extension 16 K.

J'ai sur cet appareil 5 touches F1 à F5 (+ F'1 à F'5 shiftées), dont il est dit partout qu'elles sont programmables... fort bien ! Mais il n'est dit nulle part de quelle façon et à quel usage (par une ligne de programme Type utilisant ces touches).

Est-ce de la même façon que sur certains Micropoches (TI 58C) où on peut attribuer un sous-programme ou 1 séquence de programme à ces Touches utilisateur ?

Un article à ce sujet dans "La Commode" me paraitrait le bienvenu.

Cordialement vôtre.

Alain DRIESSENS
63200 RIOM

Voir VIC à BRAC de ce numéro.

Je suis abonné depuis peu de temps à la Commode et j'apprécie beaucoup votre publication.

Je vous écris pour vous informer que l'horloge interne du VIC 20 perd son synchronisme et avance brusquement de 13 minutes lors d'une opération de chargement ou de sauvegarde sur la cassette, et d'environ 5 minutes lors d'une écriture de données sur la cassette (PRINT#1,.....).

Je pense qu'il doit se trouver une erreur dans la ROM Basic, lors de l'utilisation du 6522 pour la synchronisation de la cassette. Je n'ai pas trouvé de solution simple au problème ; j'ai signalé le fait à PROCEP qui a transmis l'information aux Etats Unis.

Si vous trouvez une astuce pour résoudre ce problème (qui est très gênant dans mes applications VIC), je vous saurais gré de m'en informer et d'en faire profiter également les lecteurs de la Commode.

Bien amicalement.

Alain AZOULAY
92260 FONTENAY AUX ROSES

Le problème que vous citez est réel, mais il ne correspond pas à une erreur. C'est simplement que le VIC utilise le même temporisateur pour la cassette et pour l'horloge temps réel et donc, il gèle l'heure pendant une opération cassette.

Une solution au problème serait d'avoir un dispositif externe qui servirait d'horloge et qu'on pourrait lire à tout moment. Il suffit d'un oscillateur (Ex 555), d'un compteur (plusieurs 7493) et d'un 74LS245 faisant la communication avec le bus de données, activé par un certain décodage d'adresses. Si une demande suffisante se manifeste, nous en commercialiserons un.

*
* *

Je possède un VIC 20, et je suis abonné à la "Commode" à partir

du numéro 3. Votre revue m'a enthousiasmé pour ses astuces et ses programmes.

Je voudrais vous poser certaines questions :
- Peut-on augmenter le nombre de colonnes du VIC ?
- Peut-on faire des "mélanges" avec les couleurs ?
- Pourriez-vous expliquer en détails la programmation en langage machine ?
En ce qui concerne les nouveautés ; un "nouveau" VIC vient de sortir :
- il a 28 Koctets de base ; pour un VIC 3,5 Koctets version de base, il suffit de rajouter quelques composants à l'intérieur.
- ce nouveau VIC a aussi, en faisant SYS 45065, la possibilité de donner avec plus de détails une erreur d'un programme.

En vous félicitant,
Amicalement,

Luc WALTERTHUM
57150 CREUTZWALD

On peut augmenter, mais légèrement, la largeur d'affichage du VIC. Voyez VIC à BRAC du numéro 5. Sinon, il existe des extensions en Angleterre qui portent l'affichage à 40 ou 80 colonnes, mais elles doublent le prix du VIC.

On ne peut pas mélanger des couleurs en un point puisqu'il n'y a qu'une case mémoire pour chaque maille où mettre le code couleur.

Nous ne connaissons pas le modèle dont vous parlez. Beaucoup de gens parlent d'un intermédiaire entre le VIC et le 64, mais il sortira plus tard.

*
* *

Bonjour, et félicitations pour vous tous, en espérant voir la "COMMODE" devenir hebdomadaire avec 200 pages.

- Par ailleurs dans le livre II de la pratique du CBM, Monsieur DAVID nous promet le livre III, oui, mais quand ?

- Dans une autre revue il a été annoncé le 500 et le 700 en remplacement du 4000 et 8000, peut on en savoir plus (capacité, CPU 8 ou 16 bits, écran 80 colonnes ? etc). Quel sera l'avenir du 9000 devant le 700 ?

- Pour empêcher un list, Monsieur DAVID préconise pour un 8032 de faire un poke 1025,0 ceci n'empêche pas la première ligne d'apparaître (sans importance un REM pourrait être placé en tête de programme) or, à la mise en marche la case mémoire 1025 est = 79, pouvez-vous m'expliquer pourquoi, lorsque l'on passe une disquette (ex l'assembleur 8000 de PROCEP, vendu sans clé donc copie possible) la case mémoire se trouve changée, et ceci sans aucune manipulation ou tentative d'un list, ce qui interdit tout LIST ultérieur.

- Dans le numéro 4, vous parliez du programme AUTORUN (existe-t-il un moyen de supprimer ceci). Peut-on rendre nos programmes AUTORUN ?

- Pensez-vous qu'il serait possible de réaliser des cours sur l'assembleur par la COMMODE ?

- Existe-t-il un club CBM en région parisienne ?

- Plus d'articles sur le 8032 ?

Dans l'espoir d'obtenir une réponse à mes questions et en vous remerciant d'avance,

Salutations,

Philippe LARICHE
92160 ANTONY

Merci de vos bons voeux pour la Commode, mais ce n'est pas demain la veille que nous aurons 200 pages. La première chose que nous ferons sera d'augmenter la densité d'impression et de passer à 96 pages.

Le volume III de la pratique du CBM est en préparation, mais Daniel Jean DAVID :

1 - cherche à incorporer des astuces sur les disques, longues à

mettre au point

2 - travaille à d'autres livres sur le VIC.

Les modèles 500 et 700 se caractérisent par de grandes tailles mémoire (128 K, 256 K et ultérieurement plus), la présence d'un ou plusieurs micros 16 bits, permettant notamment d'utiliser CP/M. Le 9000 est une machine tout à fait différente convenant surtout à l'enseignement des langages. Le 9000 ne reçoit aucun support de PROCEP alors que les 500 et 700 seront soutenus à fond.

Votre remarque sur LIST est exacte. En revanche, l'assembleur est écrit en langage machine. Il n'a donc pas besoin de la case 1025 et, de toutes façons, n'est pas justiciable du LIST BASIC.

La question des programmes AUTORUN est difficile et elle sera traitée dans le prochain numéro.

Nous lancerons dans La Commode deux séries d'initiation l'une à BASIC et l'autre sur l'assembleur.

Voyez notre section clubs et annonces. Il faut aussi bien sûr que les clubs se fassent connaître.

*
* *

Pourquoi ne pas mettre le questionnaire du feedback au dos d'une page de publicité, afin d'éviter de "Mutiler" un article de la "Commode".

D'autre part, j'aimerais connaître le mode d'emploi des touches F1....F4, F1....F4 dites "TOUCHES PROGRAMMABLES" par l'utilisateur sur le VIC 20. Est-il possible dans une prochaine parution de m'éclairer sur ce point?

Je tiens à souligner -puisque l'occasion m'en est donnée- le grand intérêt qu'ont suscité pour moi les ouvrages "La Découverte du VIC 20" et "les clés du PET/CBM tome 2" de Daniel Jean DAVID. C'est grâce à ces deux livres que j'ai pu utiliser mon VIC 20.

Vous souhaitant bon courage, je vous prie de croire, Messieurs, en mes cordiales salutations.

Alain DRIESSENS
63200 RIOM

*
* *

Pourriez-vous publier un programme de répétition automatique des touches pour CBM 40XX petit écran et qui soit compatible avec Edex. J'ai trouvé jusqu'ici quelques programmes, mais ils se plantent systématiquement si Edex est activée.

J.C. RAIMBAULT
63120 COURPIERE

On en donnera un bientôt. Il ne peut pas être compatible avec EDEX, car EDEX modifie lui aussi l'adresse d'interruption.

*
* *

Pour ceux qui trouvent le jeu SPACE INVADERS désespérément trop lent, faire POKE 3585,1 ça ira tout de suite mieux.

En effet : 3585 = 0E01(hexa)
Et on trouve en 0E00 : A9 02..

Ceci règle la temporisation d'attente au clavier.

Dans le jeu MUR DE BRIQUES
Pour avoir plus que 7 raquettes, faire POKE2311,N où N est le nombre de raquettes voulues. Si on dépasse N = 9 il y a quelques petits problèmes d'affichage, mais sans gravité.

Par contre, qui me dira comment changer les touches de contrôle (touches 4 et 6) car celles-ci commencent à souffrir sérieusement. Ceci pour les deux jeux précités.

CBM,ment vôtre,

Etienne LEGENDRE
21121 DAIX

*
* *

Au secours !!!

1 - J'ai tenté de réaliser l'extension de mon PET2001N 16K... échec ! ; toujours 15359 BYTES FREE. Le tracé du circuit imprimé était interrompu, je l'ai rétabli... toujours rien.

Mais n'y connaissant pas grand chose, je n'ai plus touché à rien. Or, ces derniers jours, surprise : dès la mise sous tension 31743 BYTES FREE ? Que s'est-il passé ; que se passe-t-il ? Quels conseils pouvez-vous me donner ? D'autant plus que maintenant je suis revenu à 16 K. Bizarre non ?

2 - J'ai une SEIKOSHA GP 100A. Bien content, sauf que je ne peux l'utiliser au mieux de ses possibilités, ne connaissant pas toutes ses instructions, surtout celles qui sont cachées (il doit y en avoir). Là encore pouvez vous m'aider ?

3 - Et là, c'est le plus important : La Commode est vraiment commode, elle nous faisait cruellement défaut. Bravo ! et merci, une revue bimestrielle serait très appréciée... trois mois, c'est long pour des affamés.

En attendant de vous lire, recevez mes sentiments les plus Cébémistes et encore merci.

P.S. Cibiste, j'espère voir un jour dans notre revue quelque chose genre RTTY? Est-ce possible ?
73.51. à toute la rédaction et 88 s à YLS.

Paul et Nicole LECLERCQ
6830 BOUILLON

1 - Il s'agit certainement d'une mauvaise soudure. Il faut vérifier soigneusement le cablage.

*
* *

Pourriez-vous venir à mon aide? Avec un ami, nous avons acquis par interêt personnel le matériel suivant :
CBM 4032, monodisque 2031 et une petite imprimante "SEIKOSHA" GP-80M d'occasion qui fonctionne bien, mais nous n'avons pas les codes de commande "BASIC" aussi je n'obtiens pas les caractères graphiques ni les caractères de "curseur programme" - Avez-vous la possibilité de me renseigner ? Comment faire ?

Merci de votre attention, je vous prie d'agréer, mes sincères salutations.

Daniel HETE
28300 MAINVILLIERS

Nous publierons dans La Commode un banc d'essai de la SEIKOSHA.

*
* *

Je suis à la recherche d'un "bon" programme d'échecs pour CBM 4032 cassette. Cependant, je n'ai jamais vu, dans les publicités et catalogues, de proposition de Sargon 2.5 qui semblait être jusqu'à ces derniers temps le meilleur programme existant pour micro-ordinateurs.

Pourriez-vous me faire savoir, cette question intéresse certainement nombre de lecteurs de votre revue, si vous connaissez des programmes d'échecs performants et le cas échéant le nom des boutiques ou sociétés qui les commercialisent.

Par ailleurs, je vous serais reconnaissant si vous pouviez me faire part de votre opinion au sujet du problème suivant :

MATERIEL :
- CBM 4032
- magnétophone à cassette stéréo Sony
- câble de raccordement ordinateur-magnétophone de la société belge B.C.M.

J'ai observé (et subi) les différents incidents de fonctionnement suivants :
à l'enregistrement :
absence des marques de début de programme ."Verify" ne trouve pas le programme à vérifier, mais trou-

ve ceux enregistrés précédemment.
.Le changement de cassette ne fournit aucune amélioration.
.Si par la suite, après avoir éteint la machine, on écrase le début du programme par un nouveau début (sans laisser s'écrire tout le programme!) la fin du programme précédent est bien trouvée (mais malheureusement, le programme n'est pas récupéré')
à la lecture :
- Enregistrement non trouvé ; parfois ordinateur "bloqué" sur "SEARCHING"
- Enregistrement trouvé, mais lecture arrêtée bien avant la fin de la lecture et indication "LOADING ERROR"
- Enregistrement trouvé, lecture faite, mais après avoir marqué "READY" le curseur ne réapparait pas et l'ordinateur est "bloqué", la seule solution consistant à l'éteindre.

Ces différents incidents qui se sont produits chacun plus d'une fois traduisent-ils une panne d'un (ou plusieurs) élément(s) et le(s)quel(s) ?

En vous remerciant par avance des réponses que vous voudrez bien me faire, je vous prie d'agréer, Messieurs, l'expression de ma considération distinguée.

Daniel DORDAIN
89290 CHAMPS SUR YONNE

1 - Le programme SARGON existe bien pour le VIC, mais nous ne l'avons pas trouvé sur CBM 4000. Nous continuons les recherches.

2 - Les incidents que vous décrivez semblent incriminer l'intensité d'écriture sur la cassette. Il y a peut être un règlage optimum à trouver pour le volume : c'est l'inconvénient de l'utilisation d'un magnéto ordinaire par rapport à celle d'un magnéto spécial Commodore où le réglage est fait une fois pour toutes. Si vous pouvez faire le montage décrit dans notre numéro 3 et comparer avec le fonctionnement du câble BCM...

*
* *

Je vous envoie cette lettre pour vous demander conseil. En effet, suite à votre article paru dans la rubrique "Vic à Brac" du numéro 4 de la Commode, j'ai effectué le branchement numéro 3. Or, seul celui-ci m'a permis d'obtenir une image et le son, mais, sans aucune couleur.

Vous remerciant par avance pour tout renseignement susceptible d'expliquer ce mystère, je vous adresse mes sincères amitiés.

Frédéric BASTART
77330 LESIGNY

L'image obtenue par la prise Péritel sur une télé SECAM est en noir et blanc. Si votre télé est PAL, il y a peut-être un problème de réglage de la couleur, à moins que votre VIC soit défectueux.

*
* *

Bien que fort béotien en informatique et pris par d'autres tâches, je prends toujours plaisir à vous lire, même si bien des subtilités m'échappent.

Pouvez-vous me communiquer l'adresse (ou lui demander de prendre contact avec moi) de Pierre THERY, car je désirerais acquérir son BASIC 3 - BASIC 4.

A ce propos, il y a près d'un an que j'attends les 5 ROMS 4.0 que j'ai payé en juillet 81 au revendeur CBM de ma région ; y a-t-il vraiment un problème chez PROCEP ou bien suis-je tombé sur un concessionnaire particulièrement incompétent ?

Bien amicalement à vous.

Jean Philippe VINCENT
49300 CHOLET

L'adresse de Pierre THERY (qui nous est demandée par de nombreux lecteurs) a été donnée dans le numéro 5.

Votre problème a été transmis à PROCEP.

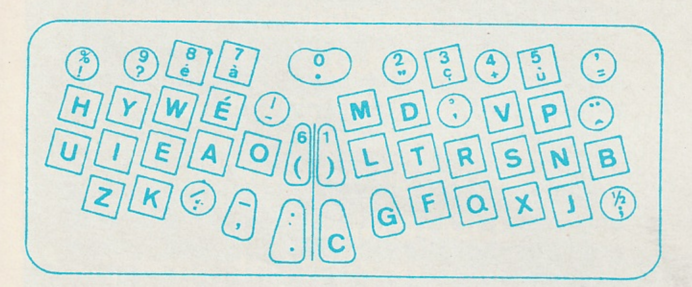
*
* *

Magazine

Clavier AZERTY

Un nouveau clavier AZERTY est apparu, dit clavier AZERTY européen, qui remplace, sur les 8000, le clavier développé par PROCEP. C'est maintenant Commodore lui-même qui prend en compte, de façon centralisée, la diversité des claviers nationaux au lieu de laisser ce soin aux différents importateurs. Rendons justice à PROCEP, d'avoir assumé le problème le temps qu'il a fallu, et d'avoir ainsi dépanné les utilisateurs. La gestion centralisée du problème des claviers est préférable : elle permet une compatibilité des différents logiciels standard avec les différents claviers. En particulier TRAITEXT II et SILICON OFFICE sont compatibles avec l'AZERTY européen.

Un français, Monsieur MARSAN a inventé un nouveau clavier français résultat de ses recherches ergonomiques (cf. ci-dessous) A quand sur les CBM ?



VIC l'ordinateur copain

Tel est le nouveau thème de promotion du VIC :

- copain de jeux : avec tous les jeux que vous connaissez, et auxquels vient récemment de s'ajouter SARGON, un des meilleurs jeux d'échecs connus actuellement.

- copain d'études : avec de nombreux programmes didactiques.

- copain pour l'initiation à l'informatique, avec le cours pratique d'autoformation

- copain pour la gestion familiale avec de nombreux programmes en cours de développement ou déjà apparus - sans oublier le programme de gestion de comptes courants, dans ce numéro.

Le VIC au Printemps

Jusqu'à la fin décembre, le VIC est présent dans la "Bulle des Jeux" au 9ème étage du Printemps - Hausmann.

Durs, durs les disques

Les deux modèles de disques durs Winchester proprement Commodore sont disponibles. Il s'agit du

9060 5 Mo 23660,70 Frs TTC et du 9090 7,5M 29590,70 Frs TTC.

Il s'agit de disques inamovibles (d'où problème de la sauvegarde) mais reliés au bus IEEE et donc de gestion totalement compatible avec celle qu'on connaît pour les disques souples.

Ce n'est pas le cas de la 8140 qui a son système propre. Mais elle a un disque fixe de 10 Mo et une cartouche amovible de 10 Mo, d'où solution facile du problème de la sauvegarde. Le prix est de 60000 F.

... et aussi les disques souples

La 8250 est disponible pour 16544,70 Frs TTC. Il s'agit d'une 8050 double face (et double tête) d'où capacité en ligne de 2 Mo au total. C'est déjà respectable. L'unité peut lire -et réécrire- les disquettes déjà formatées sur 8050.

Côté disquettes 20 cm, rien chez Commodore, mais A.I.O commercialise une unité CompuThink, compatible IBM.

Voici un résumé du prix du K octet sur les différentes unités :

2031	27 Frs
4040	30 Frs
8050	14 Frs
8250	8,25 Frs
9060	4,72 Frs
9090	4,00 Frs
8140	3,50 Frs

Système 8002

Ce système est la seconde configuration complète 8000 proposée par PROCEP, à côté du 8001. Le 8002 est formé d'une unité centrale 8096, de l'unité de disques 8250 et de l'imprimante 8023 P, pour un prix de 37550 Frs HT. Il est spécialement adapté aux logiciels TRAITEXT2, Silicon Office, PROpaie, PROcompta, PROventes...

De l'EDEX, encore de l'EDEX

Deux nouvelles versions d'EDEX ont vu le jour.

EDEX 4.4. pour 8000 avec clavier AZERTY européen.

EDEX 3.3. pour 4000. Une bonne nouvelle : contrairement à ce que nous disions dans le précédent numéro, elle fonctionne aussi avec un 4000 petit écran.

Les PDG vont bientôt offrir un VIC à leur secrétaire

En effet, nous avons pu essayer la version VIC de TRAITEXT qui fournira des fichiers lisibles pour la version 8000 de TRAITEXT (à condition de l'équiper d'une 2031).

Parmi les autres cartouches bientôt disponibles sur VIC, VICFILE (traitement de fichiers), SimpliCalc (la version VIC de VisiCalc), le langage FORTH, etc.

Concentration des imprimantes CBM

Autour de deux pôles, la 8023 à aiguilles et la 8026 B à marguerite, qui hérite du clavier de la 8026 et des possibilités graphiques de la 8027.(14054,20 F TTC)

Jouez la carte SEDERMI

La Société SEDERMI (qui édite La Commode), développe son programme de cartes d'extension du VIC. A côté de la carte bus qui permet de connecter simultanément 4 cartouches (340 F TTC), elle annonce une carte VIA (avec 2 VIA d'où 40 entrées-sorties : 690 F), une carte avec 4 supports d'EPROM(290 F TTC), une carte RAM peuplée sur option en 8, 16 ou 24 K, une carte relais, etc.

La SEDERMI se lance aussi dans la commercialisation de cassettes.

La première est une cassette utilitaire pour VIC, UTIL-VIC (voir annonce spéciale dans ce numéro).

Forum

Une exposition-forum Commodore est prévue en février-mars. Les abonnés à La Commode en seront avisés par lettre puisque cela tombe entre deux numéros.

L'Equipe de Rédaction

CONCOURS DE CREATION DE PROGRAMMES D'ENSEIGNEMENT SUR VIC 20

PROCEP lance un concours de création de programmes d'enseignement sur VIC 20, ouvert essentiellement aux enseignants familiers de la programmation sur matériel Commodore.

* Les candidats doivent adresser à PROCEP un avant projet du programme, précisant la fonction, l'objectif du programme et ses caractéristiques (mémoire nécessaire, graphique, couleurs, sons...)

* Les candidats sélectionnés pouront recevoir en prêt un VIC-20 et un lecteur de cassettes, pour une durée n'excédant pas 3 mois.

* Les programmes créés seront soumis à PROCEP pour évaluation, sur cassette accompagnée du mode d'emploi du programme. Les programmes retenus pourront être commercialisés par PROCEP, et leurs auteurs garderont alors le VIC 20 comme cadeau.

Adressez vos propositions et votre demande de prêt à

PROCEP, 21 rue Mathurin Régnier
75015 PARIS

Tél. : (23) 20.14.34

Premier Contact avec le COMMODORE 64

En attendant le dernier né de Commodore, il y a de quoi être impatient et les concurrents peuvent trembler!

Et voilà ! Le nouveau Commodore est arrivé et devrait être commercialisé véritablement en France au début de 1983. Un concurrent dangereux pour beaucoup vu son prix qui devrait être de l'ordre de 5000 francs.

A première vue

Disons tout de suite que le prix en France n'est pas encore fixé au moment où nous écrivons, mais tant aux Etats Unis qu'en Angleterre, le 64 vaut exactement le double du VIC ; la conclusion est facile à tirer.

A première vue, le 64 est un VIC, mais Commodore se mettrait très en colère si on l'appelait VIC 64. De fait, le boîtier est exactement le même que celui du VIC, à part la teinte (plus foncée) et quelques connecteurs différents. Le clavier est exactement celui du VIC.

Ce qui change, c'est l'intérieur. On a toujours une seule carte circuit imprimé -gage de fiabilité- mais elle est cette fois équipée de 64 K octets de MEV (en plus des 20 K octets de mémoire morte) : le microprocesseur est un 6510 sur lequel les seuls renseignements qu'on ait à l'heure actuelle en France sont :

- possibilité d'adressage alterné permettant de dépasser 64 K

- présence d'un port d'entrées-sorties parallèles

- même jeu d'instructions que le 6502.

D'autres circuits intégrés sont changés par rapport au VIC, sur lesquels nous reviendrons : le 6561 est remplacé par un couple 6567 + 6581. Mais déjà la taille mémoire change beaucoup de choses. A la mise sous tension, le 64 affiche

```
**** COMMODORE 64 BASIC V2****
64 K RAM SYSTEM
38911 BASIC BYTES FREE
```

ce qui est même plus qu'un CBM 8032. Il est évident qu'avec cette taille mémoire, les applications les plus élaborées et les plus professionnelles deviennent envisageables. Les langages de programmation allant au delà de BASIC deviennent possibles et Commodore a bien annoncé son intention d'en fournir.

Affichage - Couleurs - Son

Comme le VIC, le 64 gère un écran couleur, emprunté à votre téléviseur domestique. Bien sûr, les 64 livrés en France seront en PAL et il se pose les mêmes problèmes que pour le VIC. Mais ils ont la même solution : l'interface PAL-SECAM diffusé par PROCEP est d'ores et déjà disponible ; son

prix de 980 francs, un peu lourd pour le VIC, "passe" mieux, relativement à un 64.

C'est un Vidéo Interface Chip analogue au 6561, le 6567 qui gère l'affichage, suivant les mêmes principes que le VIC 20. Mais, différence fondamentale, l'affichage est maintenant de 25 lignes de 40 caractères, et la haute résolution est de 200 sur 320. Donc avec le 64, la PRINCIPALE LIMITATION DU VIC EST SUPPRIMEE. On retrouve le même format d'affichage que le P.E.T., ce qui accentue la compatibilité avec les produits antérieurs :

- même BASIC
- même format d'affichage.

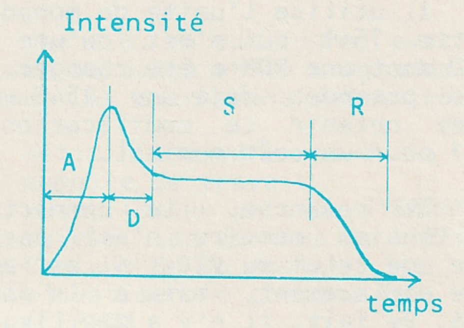
On parle même d'un package qui permettrait à un 64 de complètement émuler un P.E.T.

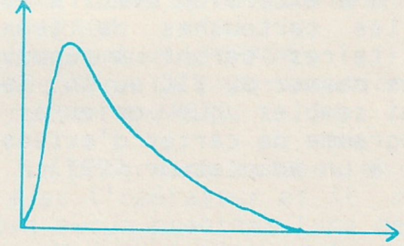
La différence de format d'affichage et la haute résolution plus fine constituent un gros morceau. Il y a cependant d'autres différences :

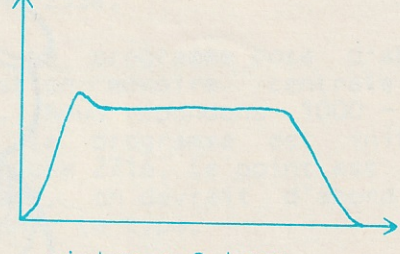
- des couleurs ont été ajoutées, en ce sens que les caractères peuvent maintenant être affichés avec les 16 mêmes couleurs que le fond de l'écran (sur VIC, il n'y en avait que 8 parmi 16). Aussi les couleurs dites "claires", orangé clair, turquoise clair, pourpre clair et jaune clair sont respectivement remplacées par brun et trois tons de gris.

- les "sprites" - Il s'agit d'un outil graphique sensationnel. Le 64 vous permet de définir des objets sur l'écran, dans une maille 21 sur 24. Vous pouvez définir jusqu'à 8 Objets. Nous traduisons le terme "sprite" (qui veut dire fantôme, esprit en anglais par objet (graphique). Nous aurions pu dire aussi "sujet" ou "forme". Ces objets sont un peu comme des caractères programmables mais plus riches (21 sur 24 au lieu de 8 sur 8) et plus faciles à gérer. C'est très facile de les déplacer sur l'écran, de les faire passer les uns derrière les autres, de gérer leurs collisions. Leurs applications sont immenses dans les jeux. mais aussi dans les affichages animés pour des programmes d'enseignement.

La partie sonore a elle aussi été très enrichie par rapport au VIC. Elle est maintenant confiée à un SID 6581, qui est un véritable synthétiseur sonore à 4 voies. Pour chaque voix, on peut décider la hauteur du son (de DO-0 à LA-7), son timbre (forme d'onde en dent de scie, en triangle, en carré ou bruit), un filtrage éventuel du son. Mais le plus important est ce qu'on appelle l'ADSR, qui détermine complètement l'enveloppe du son et distingue les instruments les uns des autres. Les 4 paramètres de l'ADSR sont programmables dans le 64 (A = attaque, vitesse de montée du son, D = descente du son à son niveau moyen, S = soutien, durée du son à son niveau moyen, et R = relaxation, durée de retour au silence, voir fig. Ci-dessous).




piano : A bref, S=0


violon : S long

L'A.D.S.R. d'un son

La qualité du son du 64 justifie la possibilité de le relier à une chaîne hifi.

Périphériques

Le 64 admet certains périphériques en commun avec le VIC. Il a deux connecteurs pour poignées de jeux au lieu d'un.

Il a le même standard IEEE série que le VIC et il utilise la même imprimante dérivée de la Seiko 80 que le VIC. On peut espérer qu'il pourra bientôt utiliser une imprimante dérivée de la Seiko 100.

Il utilise l'unité de monodisquette 1541 qui n'est qu'une VIC 1540 dont une ROM a été changée. Si vous possédez déjà une 1540 vous devez obtenir la modification en 1541 pour un coût nominal.

En revanche, le connecteur d'extension mémoire n'est pas le même que celui du VIC : il est au même emplacement, mais a le pas 2,54. De fait, il n'y a pas lieu de mettre une extension mémoire au 64, mais les cartouches de jeux ou d'utilitaires seront à renouveler si vous passez du VIC au 64. Sedermi, qui semblez vous orienter vers un programme de cartes d'extension, pensez à un adaptateur (???).

Conclusion

Avec le 64, Commodore nous semble avoir frappé un grand coup et, peut-être détenir enfin la machine personnelle que tous attendaient.

Le 64 est tout de même deux fois plus cher que le VIC, donc celui-ci doit pouvoir conserver son marché.

En tous cas, si vous comptiez acheter un VIC et l'équiper d'une extension d'affichage 40/80 qui double son prix, pensez au 64 !

Si l'on compare le PET 2001 (vendu environ 7000 francs à sa sortie en 77) et le 64 qui revient à 7000 Frs aujourd'hui avec une petite télé couleur, on mesure les progrès accomplis :

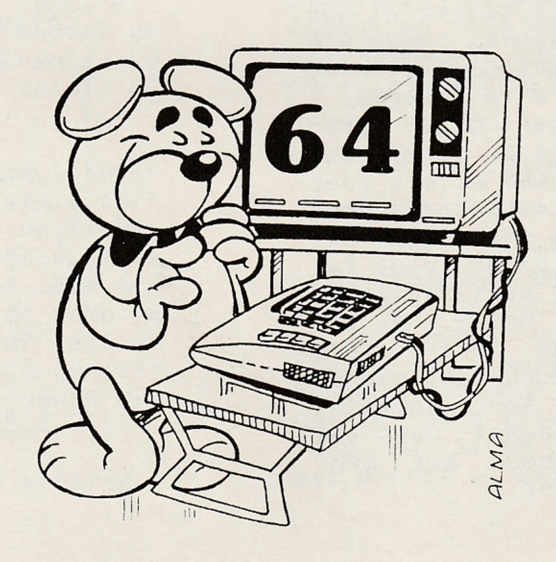
- taille mémoire passée de 8 K à 64K

- affichage couleur et haute résolution déjà présente

- les effets sonores.

Progrès qui auront pu être suivis pas-à-pas, grâce à la compatibilité des BASIC. Sur la route de ces progrès le VIC 20 aura constitué une étape intéressante et sympathique.

Pierre-Etienne THALBERG



Banc d'essai de deux Produits Chatain-Blanchon

CHATAIN-BLANCHON est une véritable mine pour les commodoristes. Leur publicité dans notre numéro 5 vous montre l'ampleur de la liste des produits qu'ils font pour les Commodore : beaucoup sont importés d'Allemagne, pays où les Commodore sont plus nombreux qu'en France. Nous analysons dans cet article deux de ces produits parmi les plus significatifs : un programmateur d'EPROM à 1000 Francs H.T. et une ROM concurrente d'EDEX. Enfin, n'oubliez pas que C.B. se charge de certaines opérations sur CBM, notamment des augmentations de taille mémoire ; pensez-y si vous n'êtes pas très bricoleur.

Le programmateur d'EPROM C.B.V4

Un programmateur d'EPROM à 1000 frs hors taxe, il y a de quoi rêver ! Même sans aller jusqu'à la solution industrielle RACK SYSMOD + carte VIA +PROSYS d'Eristel, toutes les autres solutions proposées sur le marché sont beaucoup plus chères.

Le programmateur C.B. V4 est bien présenté dans un coffret plastique de 6 x 11 x 19 cm, sans luxe inutile. Il est relié au secteur par un còrdon et au connecteur port utilisateur par un câble en nappe: le cordon et le câble en nappe sont, à notre avis, un peu courts. Le dessus du coffret est occupé par un interrupteur à glissière qui permet de mettre ou enlever la tension de programmation de 25 V (celle-ci n'est nécessaire que pour les opérations de programmation proprement dites, les autres opérations comme lecture ou vérification n'en ont pas besoin), et un support à force d'insertion nulle pour l'EPROM. Bravo à Chatain-Blanchon de n'avoir pas lésiné sur ce dernier point qui est essentiel.

Le C.B. V4 est capable de programmer toutes les EPROM mono-tension du marché, c'est-à-dire 2758, 2508(1K x 8), 2516, 2716(2Kx8) et 2532, 2732 (4K x 8). Les 2732 nécessitent un petit câblage d'adaptation expliqué dans la notice. Les plus pratiques pour les Commodoristes sont les 2532 qui peuvent remplacer une ROM du CBM.

Le C.B. V4 est fourni avec une notice adéquate et un programme qui permet les opérations suivantes :

M : définir les adresses de la mémoire de travail
S : sélectionner le type d'EPROM
R : relever les données de l'EPROM (en vue de sa copie)
D : visualiser le contenu de l'EPROM
C : contrôler la virginité de l'EPROM
P : programmer l'EPROM
V : vérifier la programmation
E : retour à BASIC.

Le programme marche en dialogue avec l'opérateur et il ne nous a posé aucun problème lors de notre essai. Voici à titre d'exemple le dialogue qui a lieu pour la copie d'une 2532.

Le programme pose d'abord la question mémoire centrale début fin. Nous répondons 1000 - 17 FF car le programme est initialisé pour une 2716. Ce point est un peu gênant, on devrait d'abord sélectionner le type.

Nous émettons alors une commande S pour passer en 2532 : on répond 4 à la question nombre de K octets. Il faut alors redéfinir la mémoire centrale en émettant une commande M et en répondant 1000-1FFF à la question début - fin.

Nous plaçons alors la REPROM à copier sur le support, et la vérrouillons avec le levier. Une commande R va la copier en mémoire centrale.

Ensuite nous plaçons la REPROM vierge à programmer sur le support. Une commande C permet de vérifier sa virginité. Nous émettons alors la commande P. A la question DE, nous répondons 0000 pour dire que nous voulons programmer l'EPROM depuis le début. L'écran nous indique alors de mettre le 25 V sous tension. Lorsque c'est fait, on appuie sur une touche quelconque du CBM et la programmation démarre. Elle dure un peu moins de 4 minutes.

Nous obéissons à un message spécifiant d'enlever le 25 V (appui sur une touche quelconque quand c'est fait) et c'est terminé. Pas tout à fait ! Il est prudent d'exécuter une commande V de vérification (nécessaire seulement pour les 4 K octets). Nous avons pu vérifier que l'EPROM-copie fonctionnait:elle avait été correctement programmée.

Bien sûr, si au lieu d'une commande R on avait lu un fichier dans la mémoire centrale, on aurait pu transformer ce fichier en ROM.

CONCLUSION :

Ce programmateur de REPROM fonctionne parfaitement ; il offre toutes les fonctions voulues pour ce genre d'applications et ce, pour un prix qui le met à la portée de l'amateur. La meilleure preuve en est que, au lieu de rendre le programmateur après l'essai, nous l'avons acheté ! Il est possible que C.B. fasse une version VIC de ce programmateur.

La ROM SUPER-BASIC

Il s'agit d'une ROM (en fait une EPROM) qui se met sur l'un des supports libres du CBM, et ajoute des instructions à BASIC. Elle existe en versions pour 2000, 3000, 4000 et 8000, avec choix de l'adresse de départ en $9000, $A000 ou $B000 (les deux premiers choix seulement pour 4000/8000). Nous avons essayé une version 3000 en $A000 (pour coexister avec EDEX). Elle vaut 490 frs H.T.

Toutes ces ROM d'extension de BASIC s'adressent à peu près au même type de problèmes et il y a un certain recouvrement entre elles. Toutefois, EDEX et SUPERBASIC ne font pas totalement double emploi et nous avons vérifié que les deux pouvaient coexister.

Nous passons maintenant en revue les commandes de SUPERBASIC en signalant celles qui sont couvertes aussi par EDEX. Une grande originalité de SUPERBASIC est de présenter des commandes de lecture écriture accélérées sur cassette (dans un rapport 8), tout en utilisant le magnéto habituel. Ces commandes seront étudiées à la fin. Dans ce qui suit, (E) signifie "il y a quelque chose d'analogue dans EDEX".

EXTENSIONS DE BASIC

!CALL : appel de sous-programme mais l'étiquette peut être calculée comme expression arithmétique ou être une chaîne de caractères entre guillemets : on sautera à la première ligne ayant cette chaine en REM.

Exemple : si l'on a 400 REM "ZOZO" !CALL 4 * 100 et !CALL "ZOZO" sont équivalents à GOSUB 400.

(E) !ELSE : permet le IF THEN ELSE mais !ELSE doit être en tête de ligne, l'exemple donné dans la notice est faux (fâcheux !) Voici un exemple qui marche :

```
10 INPUTA
20 IFA=0THEN PRINT"O"
30 !ELSE PRINT"N"
40 PRINT"A"
```

!INST (A,A$,B$,C) : met dans A la position de la sous-chaîne B$ dans A$ en cherchant à partir de la position C.

Exemple :

A$="BONJOUR": B$="JOUR" : C=2
!INST(A,A$,B$,C) donnera à A la

valeur 4. Si la sous-chaîne est absente ou si C a une valeur supérieure ou égale à la position de départ, on obtient A=0. Ceci est mal précisé par la notice, ainsi que le fait que tous les paramètres doivent être des variables.

!JUMP : comme !CALL sauf que c'est GOTO qui est remplacé. La notice oublie de préciser que les instructions de SUPERBASIC doivent être en tête d'instruction, donc en tête de ligne ou précédées par deux points :
IF A=0 THEN !JUMP "FIN" ne marche pas. Il faut écrire
IF A=0 THEN :!JUMP "FIN".

Sinon, cette possibilité d'utiliser les labels aussi parlants que l'on veut est un gros avantage de SUPERBASIC.

(E) !TYPE(D),N : imprime le nombre N avec D décimales (avec arrondi).

Ex : TYPE(5),1.23456789 imprime 1.23457.

C'est moins riche que PRINT USING mais couvre l'essentiel en étant plus simple. Les exemples de la notice écrivent !TYP ce qui fait SYNTAX ERROR.

EXTENSIONS DIVERSES (moins essentielles que les précédentes).

(E) !BEEP X,Y : produit pendant Y millisecondes un son de fréquence inversement proportionnelle à X(0<X<255).

!HALT X : fait une pause de X centièmes de seconde.

!HIMEM $XXXX : réserve pour la protéger de BASIC la zone mémoire qui commence en $XXXX.

Ex: !HIMEM $7000:CLR

(E) !KOPIE : copie l'écran sur l'imprimante.

!RESET : desactive SUPERBASIC (manque cruellement dans EDEX).

!SETZE A : permet un RESTORE sélectif sur la ligne de DATA de numéro indiqué par A (constante ou variable).

!VERT(A) : saute A lignes. Les ( ) sont nécessaires : les syntaxes ne sont pas trés homogènes.

COMMANDES FACILITANT L'UTILISATION DE L'ASSEMBLEUR

SUPERBASIC contient un sous-ensemble d'EXTRAMON en ce sens qu'il contient un mini-assembleur et un mini-désassembleur.

!ASS adresse : assemble à partir de l'adresse spécifiée en décimal ou en hexadécimal si elle est de la forme $AAAA. Le système fournit les adresses et vous tapez votre code en symbolique.

Exemple :

```
!ASS $1000
$1000 : LDA #$41
$1002 : JSR $FFD2
$1005 : RTS
$1006 : (retour chariot pour
          arrêter)
```

!CHANGE valeur : conversion. Si valeur est donnée sous la forme nnnnn, elle est convertie en hexa. Si elle est donnée sous la forme $xxxx, elle est convertie en décimal. Ces conversions manquent cruellement à BASIC standard, SUPERBASIC comble la lacune.

!DIS adresse : désassemble 12 instructions. On désassemble les 12 suivantes à chaque appui sur SPACE. On arrête par STOP.

A noter, ce qui est bien pratique tant pour !DIS que pour !ASS, que si l'adresse est fournie en hexa, les écritures se font en **hexa, de même pour le décimal.**

!DO adresse, a,x,y : lance le programme en langage machine à partir de l'adresse indiquée. Les arguments facultatifs a, x et y iront au départ dans les registres de même nom.

Exemple : !DO 4096, 0, $F3, Y

!TABLE adresse: permet de créer une table de texte en mémoire.

Exemple :

```
    TABLE $033A
    $033A : "BONJOUR
    $0341 : retour chariot
```

met à partir de 33A les valeurs $42, 4F, 4E, 4A, 4F,55 et 52.

COMMANDES PERIPHERIQUES

!BILD "nom" charge le fichier nom dans la mémoire d'écran

!OUT "d:nom" vide la mémoire d'écran sur disque (sauvegarde d'image)

!WRITE "d:nom", départ, fin, périph : sauvegarde sur périphérique mais en précisant les adresses de début et de fin comme dans la commande .S du moniteur. !OUT est un !WRITE avec départ =$8000, FIN = $8800 et périph = 8.

Commandes cassette rapide

Super Basic gère le magnéto Commodore à 2500 baud au lieu de 300 comme le logiciel standard.

!HOL lit dans la première variable du programme, les données écrites par !PUT1 ou !PUT2.

!PUT1 ou !PUT2 : écrit sur le magnéto 1 (resp. 2) les données de la dernière chaîne de caractères définie.

!RIN "nom" : charge le fichier "nom" sans modifier les pointeurs, donc en overlay.

!ROUT "nom", XXXX,YYYY : vide sur le fichier nom, en vue d'une lecture par !RIN, le contenu de la mémoire entre XXXX et YYYY (hexa sans signe $).

!SLOAD et !SSAVE comme LOAD et SAVE, mais 8 fois et demie plus vite.

Exemple : !SSAVE "ZOZO",2

Conclusion

Voici une belle brochette de nouvelles commandes pour votre BASIC. Certains manques de EDEX sont comblés, d'autres commandes originales apparaissent, notamment le mini assembleur et la gestion des images écran sur disque. Certains points de détail de la notice sont à revoir. En tous cas, un achat qui vaut le coup.

Pierre Etienne THALBERG

Qui a dit que basic n'était pas structuré ? (fin)

Suite du numéro 5 ...
... ou comment écrire correctement de bons programmes

AUTRES INSTRUCTIONS

D'autres instructions sont apportées par des additions dans le BASIC du CBM. EDEX doit être cité en tête, car il est fourni en série sur les CBM 8000. Il existe d'autres additions facilitant les opérations sur disque, en traitement IEEE, etc. Disons que toutes les instructions spécialisées, si elles facilitent la programmation, restent figées au système utilisé. On considèrera que le langage fait partie intégrante du système choisi, sans pour autant condamner l'absence de transportabilité : la puissance apportée par telle ou telle addition spécialisée compense le côté fermé.

AUTO est très pratique à l'édition. On numérote généralement de 10 en 10 les lignes pour permettre d'insérer facilement des lignes oubliées. Une petite règle peut être dite à ce sujet : si l'on doit rajouter à un moment quelconque plus de 9 lignes entre deux lignes BASIC, c'est que l'étude a été mal faite ! On préfèrera insérer un GOSUB plutôt que de renuméroter tout...

CALL, malgré sa puissance, appelle les mêmes remarques que SYS. A réserver pour les applications "pointues"; on l'utilisera de préférence à USR, car elle est plus puissante.

DUMP ne liste que les variables qui ont été utilisées pendant l'exécution précédente. Très pratique au DEBUG, elle ne peut pas servir à faire un catalogue de TOUTES les variables, d'autant qu'elle ignore les variables indicées.

FIND devrait servir à outrance au moment de l'édition. Un article est d'ailleurs déjà paru à ce sujet. Son intérêt devient immense pour rechercher la fonction de certaines variables si l'on a eu la précaution de les nommer avec méthode. Une bonne précaution compense le fait qu'on peut déclarer ses variables à tout moment : en faisant un FIND avant de déclarer une variable, on sait tout de suite si elle existe déjà. Cela paraît une vérité de La Palisse, mais moins d'erreurs se produiraient si l'on usait de cette astuce systématiquement.

ELSE a été abordé à l'occasion de THEN. Insistons sur le fait qu'il aide le programmeur à structurer son organigramme... quand il en fait un !

MERGE permet d'appeler des blocs d'exécution sans perdre les variables en cours. C'est une bonne occasion d'augmenter la structure de l'ensemble. Cette forme d'utilisation se justifie dans les applications de gestion qui sont souvent très lourdes.

PRINTUSING, dans le CBM, accepte autant de formats que l'on veut. Comme DEF FN, DIM et autres déclarations de ce genre, on ne saurait que trop répéter que la déclaration des formats gagne, pour la clarté, à être faite en début de programme.

RENU est une cause de dilemnes chez les programmeurs. Le fait que le numéro de ligne serve de label dans le langage BASIC fait qu'il a une signification figée. Si l'on sait par exemple que tel sous-programme débute en 1000, on éditera instinctivement GOSUB 1000 chaque fois que l'on a besoin de lui. Encore une vérité de La Palisse qui n'en est plus une si l'on renumérote le programme. Il faut alors lister le programme et aller à la pêche aux nombres pour se constituer une nouvelle table des labels. Disons, pour contenter tout le monde, qu'il est dangereux de renuméroter son programme en cours d'élaboration, mais qu'il est élégant de le présenter bien tout-à-fait terminé un jour.

SWAP est la consécration du MERGE. Elle permet d'augmenter virtuellement la taille du programme utilisateur en n'ayant en mémoire centrale que le bloc actif. Elle force l'analyse poussée du programme, donc sa structure. Tant mieux !

UNTIL et WHILE, à l'instar de PASCAL, sont apportés par l'addition COMAL. Cette addition n'étant malheureusement pas assez répandue, ces instructions doivent rester à l'usage de quelques initiés qui seront, en plus, pénalisés par le fait que leurs programmes sont intransportables.

Les branchements dits labellés permettent de décrire directement en clair l'endroit ou l'on veut aller. Cette possibilité est offerte sur le nouveau produit industriel SYSMOD-200, et l'expérience montre qu'on s'y habitue très vite! Cet agrément se double du fait que la renumérotation redevient un argument positif. Souhaitons que cette facilité devienne courante dans l'avenir.

LES MAUVAISES HABITUDES

Elles se situent à tous les niveaux de l'élaboration d'un programme. Il m'est d'autant plus facile d'en parler que, comme tout le monde, je les ai eues, ces mauvaises habitudes !!! C'est la nécessité de sortir du travail efficace (pour autant qu'on puisse le garantir) qui amène le programmeur à procéder par ordre...

1 - L'ANALYSE

On dit souvent qu'un problème bien posé est à moitié résolu. C'est primordial dans la programmation. Une mauvaise habitude consiste à foncer à partir d'idées préconçues sur un problème à peine énoncé. Ces idées peuvent être très bonnes, mais on a tendance à se fixer sur certains points qui ne sont pas essentiels dans l'exécution globale ; le programme devient alors rapidement mal articulé et dur à mettre en oeuvre. Evidemment, on ne peut que conseiller de faire des organigrammes, ce qui amène les réflexions suivantes.

Personne n'aime faire d'organigrammes, mais on peut sérier les problèmes sans pour autant se conformer à une telle rigueur de procédure. Il suffit de partir du cahier des charges en faisant ce qu'on appelle une analyse descendante. Cela ne nécessite pas toujours beaucoup de temps d'écriture; le but est d'éclater le problème général en une imbrication de modules qui, chacun pris séparément, soit ne pose plus qu'un problème mineur pour la mise en oeuvre, soit est susceptible de se décomposer à son tour de la même façon. De grandes théories ont été élaborées sur cet éclatement des modules en distinguant la routine principale, les sous-routines principales, les sous-programmes principaux, etc., qui donnent à l'imbrication générale des modules un aspect hiérarchique tout ce qu'il y a de militaire. Le programmeur professionnel doit avant tout travailler vite ; comme le physicien qui ne se préoccupe jamais de savoir si la fonc-

tion qui l'intéresse est continue avant de la différencier, le programmeur va laisser le fond de la théorie de côté pour n'en garder que les lignes principales présentés à l'esprit.

On distinguera l'exécution principale dont le nom n'est autre que le titre de l'application elle-même. L'enchaînement des actions qui doivent être assurées se présente comme autant de modules représentés par des "pavés" qui seront analysés en leur temps. Cette exécution principale est unique, ou rebouclée sur elle-même avec une porte de sortie en fin d'exécution globale. Notons qu'une initialisation est toujours nécessaire et qu'il est bon de considérer le programme global comme un gigantesque sous-programme que l'on pourra toujours appeler indépendamment dans le cadre d'une extension future de l'application. Chaque module, s'il a bien été distingué des autres, doit être étudié de la même façon, à un échelon d'imbrication supplémentaire près. Enfin, tous ces modules peuvent faire appel à de mêmes sous-programmes élémentaires qui reviennent souvent dans l'exécution globale. C'est ce qui se produit dans le cas d'entrées-sorties qu'elles soient industrielles ou simplement des fonctions d'affichage, stockage sur disque, etc.

En principe, chaque module peut alors être considéré séparément. Pour de gros programmes, une équipe de travail va se répartir les tâches selon les spécialisations de chacun. Le "mariage" entre chaque module développé séparément est possible après déclaration commune des variables et adresses utilisées, ce qui nous ramène à insister sur la méticulosité avec laquelle il convient d'utiliser la trop facile attribution des variables en BASIC. Dans un travail en équipe, la déclaration d'une nouvelle variable en cours d'élaboration doit se faire au su de tout le monde. L'intérêt d'une telle méthode est qu'un module spécialisé bien conçu peut être réutilisé dans une autre application par la suite ; le temps de développement se trouve non seulement amorti, mais fructifiant.

2 - LA PROGRAMMATION

Elle peut se faire à partir de simples listes d'exécutions élémentaires au sein d'un module. L'inconvénient est que c'est difficillement transportable au point de vue langage ; on ne programme pas une boucle de la même façon en BASIC et en assembleur, de même que PASCAL offre des possibilités supplémentaires avec WHILE et UNTIL. Par contre, un programmeur entraîné ira souvent très vite à la mise en route... en constatant que les "bugs" sont toujours aux endroits les plus simples à développer !

Il est bon d'attacher un soin tout particulier à l'édition, en détachant bien les fonctions (REM est là pour cela, servons-nous en). Les commentaires sont évidemment indispensables et il faut faire effort sur soi-même pour ne pas se laisser entraîner par une certaine fougue à l'édition au détriment de la compréhension. Le gain de temps et d'efficacité se retrouvera au "DEBUG". Une relecture systématique de la ligne que l'on vient d'éditer force à se concentrer sur l'action qui doit en découler et permet souvent, (outre la rectification des fautes de frappe), de déceler une erreur de conception avant la première mise en route.

S'il est mauvais pour la santé de passer le plus sombre de ses nuits devant le petit écran vert, il est par contre nécessaire de faire une pause régulièrement à la fréquence qui convient pour chacun: l'expérience prouve que les bugs se retrouvent en "essaims" au moment où la fatigue s'est manifestée (surtout la fatigue visuelle que l'on combat efficacement par des protections anti-reflets et/ou le port de verres filtrants).

3 - LES ASTUCES

S'il y a une chose que le pro-

grammeur adore, c'est bien les astuces. Plus il fait des choses incompréhensibles pour ses voisins, plus il jongle pour gagner quelques octets et plus il est content.

Il faut bien se dire qu'arrivé au niveau professionnel, les astuces ne se justifient que pour gagner du temps ou de la place à l'exécution. De même pour les adresses "secrètes" d'un système accédées par un PEEK ou autre WAIT; si elles sont bien pratiques par l'apparente simplicité d'utilisation et la puissance à l'exécution qu'elles apportent, elles n'en sont pas moins dangereuses dans le fait qu'elles sont liées impérativement à une machine unique. Rappelons à nouveau les problèmes posés quand le PET à été remplacé par le CBM 3032, puis par le 4000/8000. Au sein d'une même marque, le langage évolué (BASIC) est resté compatible, et c'est un bon point, mais le noyau de fonctionnement de chaque système présente des variations. Celles-ci sont inévitables car inhérentes à la vocation de chaque machine ; par contre, se servir des astuces liées à ces particularités sort du cadre d'une utilisation générale.

Cela nous amène tout naturellement à réfléchir sur le choix du langage pour une application. BASIC permet une approche facile mais limitée au point de vue de la puissance, il faut l'accepter et n'utiliser les accès internes à la machine que pour des applications très spécifiques. En voici quelques exemples : 1) dans un jeu animé, on a besoin de faire un affichage rapide ou calculé, on admettra de se servir de PEEK et POKE pour accéder à l'écran ; 2) dans une application industrielle, une entrée/sortie sera également pilotée par une lecture/écriture directe en mémoire centrale du processeur ; 3) par contre, on s'interdira d'accéder à la mémoire interne pour modifier le fonctionnement propre, il s'agit d'astuces qui sont incompatibles avec la philosophie du langage BASIC, réservons cela à la programmation-système en langage assembleur. Dans tous les cas, une intervention sur les adresses secrètes doit se faire en sous-programme documenté facilement accessible pour d'éventuelles modifications ; ces sous-programmes doivent être considérés comme des utilitaires indépendants de BASIC dans leur fonctionnement.

Les astuces de conception de programme sont admissibles si elles sont correctement commentées. Voici un exemple pratique qui pourra servir dans des jeux (et aussi pour des affichages sérieux animés !) quand il s'agit de commander des déplacements dans les quatre directions. Il est courant de choisir les touches 2, 4, 6 et 8 du clavier numérique pour demander d'aller respectivement en bas, à gauche, à droite ou en haut. Pour se brancher vers l'un des quatre sous-programmes de déplacements, une astuce permet de se servir directement de l'instruction ON...GOSUB :

```
XX GET R$ : IF R$ = "" THEN XX:
REM PREND UN CARACTERE
YY U = VAL(R$)/2 : REM POUR
ANALYSE 1 A 4
ZZ IF U<1 OR U>4 THEN XX :
REM NON VALIDE
TT ON U GOSUB AAA,BBB,CCC,DDD
```

4. LA MISE AU POINT

C'est maintenant que les bonnes habitudes doivent porter leurs fruits. Chacun sait qu'un programme ne marche JAMAIS du premier coup, il est donc indispensable de pouvoir accéder facilement à ce qui vient d'être édité pour le mettre au point. Tout commentaire, déclaration bien placée et autre indication devient la bienvenue pour éviter de passer des heures à comprendre ce qui marche mal.

Il est difficile de donner des règles générales pour le "DEBUG". Disons, pour commencer qu'il y a deux types de méthodes dont l'efficacité dépend uniquement de la pratique du programmeur. On distingue "l'école statique" et "l'école dynamique" : de bien grands mots pour dire que certains sont plus à l'aise en faisant une analyse ré-

tro-active sur le listage (listing en anglais), et d'autres préfèrent analyser les erreurs sur le terrain en exécutant le programme là où il "se plante". Que chacun reconnaisse en toute honnêteté que les deux méthodes doivent en fait se concilier.

La méthode statique consiste à analyser sur listage les modifications que doivent supporter les variables et essayer de faire le rapprochement avec les défauts constatés en fin d'exécution du module mis en cause. Elle est particulièrement efficace pour piéger le "bug" le plus classique qui consiste à déclarer deux fois la même variable pour des actions différentes. Ceci est impossible si l'on déclare convenablement les variables avec le rôle qu'elles doivent jouer ; cela se fait rarement en BASIC, mais un recensement effectué sur listage dans le module incriminé doit permettre le dépistage rapidement. A ce sujet, l'addition DUMP au BASIC est la bienvenue.

La méthode dynamique consiste à regarder "en temps réel" les variations que subissent les protagonistes pendant l'exécution. La fonction STOP est la première qui permet ce DEBUG ; on fixe avec STOP un point d'arrêt, on regarde les états à ce point, on fixe un point plus loin et ainsi de suite et l'on trouve ainsi la zone ou les défauts apparaîssent. A la manière d'une enquête policière, le noeud se resserre autour de la "verrue"... que l'on condamnera sans pitié ! TRACE et DUMP seront appréciés pour cette démarche qui se prête très bien à la compréhension de la modification d'une chaîne de caractères calculée, par exemple.

En principe, les défauts de séquencement, s'ils ne sont pas dus à une interversion ou un oubli d'appel de sous-programme dans la séquence principale, viennent soit d'un défaut d'analyse, soit de l'insuffisance du système ; dans les deux cas, il faudra tout reprendre à zéro.

Notons qu'un très grand danger peut se présenter si l'on néglige de faire tourner le programme dans les conditions les plus diverses. C'est bien sûr dans la configuration que l'on n'aura pas testée qu'il tombera en panne. Si ce n'est pas fait au développement, l'utilisateur final s'empressera de se mettre dans les conditions optimales pour trouver un "bug" qui était théoriquement impossible.

5 - LA MAINTENANCE

C'est l'aspect le moins intéressant de la profession : développer un programme est passionnant, le dépanner deux mois plus tard ou le modifier un an après n'a rien d'excitant ! C'est pourtant dans cette phase qu'on apprend le plus : c'est en faisant des faux pas qu'on évolue.

Dans cette phase, tout peut arriver, même la réécriture totale du programme... On apprécie d'autant plus la présence de documents descriptifs clairs. Il arrive fréquemment qu'une autre personne que l'auteur doive intervenir ; c'est dans ces cas-là que tout aspect astucieux de la programmation est négatif. Au sein d'une équipe, on retiendra que les programmes doivent être le plus neutres et le plus documentés possibles pour garder leur efficacité, quitte à perdre en taille mémoire : financièrement parlant, la mémoire coûte moins cher que le temps de programmation.

Alors, on peut se demander à quels particuliers s'adresse ce paragraphe. Au moins à ceux qui font de la programmation une discipline sérieuse et non une acrobatie irréfléchie (il n'y a pas de filet en informatique). A tout moment, un programme individuel peut être modifié par son auteur ou quelqu'un de sa famille ; il est donc nécessaire de préparer au départ les éléments indispensables à cette maintenance.

CONCLUSION

Dans tous les domaines, le fait de bénéficier d'améliorations tend à pousser à la facilité. L'apparition du langage BASIC en est un exemple. Y céder dans un premier temps est bien naturel, mais s'y enterrer définitivement n'est certainement pas constructif. La réflexion sur une certaine manière de se servir de ces facilités doit apporter une dimension supplémentaire à l'usage qu'on en fait.

Une question, si vous êtes atteint de programmite (dangereuse intoxication de l'informatique), dans laquelle des trois catégories narrées dans l'avant-propos vous classez-vous ?

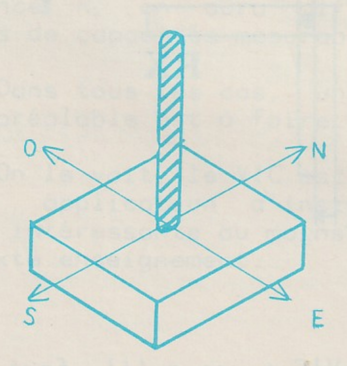
François VULQUIN

Périphériques de Jeux sur le VIC

Nous explorons aujourd'hui le connecteur latéral du VIC consacré aux jeux.

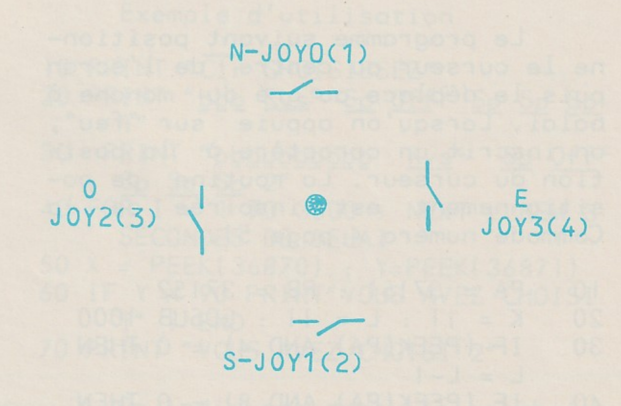
POIGNEES DE JEU

Il y a deux sortes de poignées de jeu, les manches à balai et les raquettes. Les deux possèdent un levier susceptible d'être orienté dans toutes les directions (fig. ci-dessous).



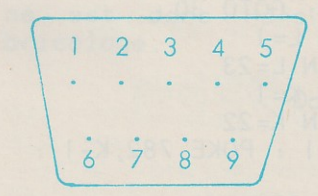
Il y a en général un bouton en bout du levier (le bouton de "tir") qui commande un interrupteur.

Les manches à balai possèdent (outre le bouton "feu"), quatre interrupteurs obéissant au schéma ci-après, qui indique aussi la connexion standard au connecteur. Le bouton "feu" est relié à l'entrée "crayon lumineux" (broche 6).



CONNECTEUR DE JEUX

vu de l'extérieur du VIC



Broche	Rôle	Borne de VIA
1	JOY 0	PA 2 *
2	JOY 1	PA 3
3	JOY 2	PA 4
4	JOY 3	PB 7 *
5	POT Y	6561 br. 18
6	cr.lumin.	6561 br.37 etPA5
7	+5V	-
8	masse	-
9	POT X	6561 br. 17

* PA = adresse 37151 ;
PB = adresse 37152

Le tableau suivant indique les interrupteurs fermés en fonction de l'indication du levier, ainsi que le brochage du connecteur.

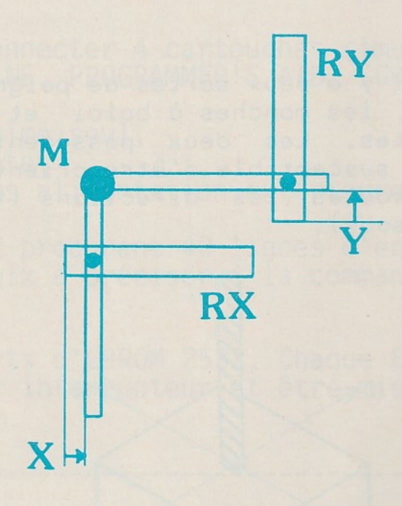
Direction	Interrupteurs fermés	TEST (inter. fermé = broche de VIA à 0)
N	JOY0	IF PEEK(PA) AND 4 ...
N-O	JOY0-JOY2	IF PEEK(PA) AND 20 ...
O	JOY2	IF PEEK(PA) AND 16 ...
S-O	JOY1-JOY2	IF PEEK(PA) AND 24 ...
S	JOY1	IF PEEK(PA) AND 8 ...
S-E	JOY1-JOY3	IF(PEEK(PA)AND8)+(PEEK(PB)AND32)...
E	JOY3	IF PEEK(PB) AND 32 ...
N-E	JOY0-JOY3	IF(PEEK(PA)AND4)+(PEEK(PB)AND32)...
"feu"	cr. lumin.	IF PEEK(PA) AND 32 ...

Le programme suivant positionne le curseur au centre de l'écran puis le déplace au gré du manche à balai. Lorsqu'on appuie sur "feu", on inscrit un caractère à la position du curseur. La routine de positionnement est inspirée de la Commode numéro 4 page 51.

```
10   PA = 37151 : PB = 37152
20   K = 11 : L = 11 : GOSUB 1000
30   IF (PEEK(PA) AND 4) = 0 THEN
     L = L-1
40   IF (PEEK(PA) AND 8) = 0 THEN
     L = L+1
50   IF (PEEK(PA) AND 16) = 0 THEN
     K = K-1
60   IF (PEEK(PB) AND 32) = 0 THEN
     K = K+1
70   IF (PEEK(PA) AND 32) = 0 THEN
     PRINT CHR$(230)CHR$(157);
80   GOSUB 1000 : GOTO 30
1000 IF L<1 THEN L=1
1010 IF L>23 THEN L=23
1020 IF K<1 THEN K=1
1030 IF K>22 THEN K=22
1040 POKE 781,L-1 : POKE 782,K-1 :
     POKE 783,0
1050 SYS 65520 : RETURN
```

LES RAQUETTES

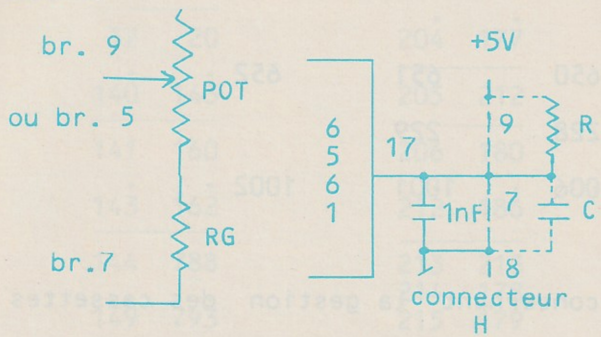
L'inconvénient des manches à balai est qu'ils ne "décèlent" que 8 directions. Les raquettes peuvent, elles, mesurer des variations continues. Une raquette est formée (figure ci-après) de deux potentiomètres à curseur linéaire RX et RY, actionnés par les glissières perpendiculaires liées à la manette M. Les positions X et Y des curseurs sont proportionnelles aux résistances RX et RY et aux coordonnées de la manette M. Pour connaître les coordonnées de la manette M, il suffit donc de mesurer les résistances RX et RY.



Le VIC a ce qu'il faut pour cela. Si on relie une résistance entre la broche POTX (ou POTY) et la broche + 5V du connecteur H, on obtient en 36872 (X) ou 36873(Y) une valeur sur un octet (donc comprise entre 0 et 255. Il faut étalonner pour chaque VIC, mais, en gros, il faut utiliser un potentiomètre de valeur maximum dans les 200-250 kohms. La valeur 0 est nuisible. Pour l'éviter on conseille le montage ci-dessous, avec une résistance de garde RG de 3 kohms. La valeur obtenue par PEEK(36872) ou PEEK (36873) est, en gros, le nombre de kohms de la résistance.

APPLICATION A LA MESURE DES RESISTANCES ET DES CAPACITES.

Le montage réalisé dans le VIC est le suivant (on prend le cas de X dans la figure ci-après). On constitue un circuit RC avec le condensateur de 1nF, déjà présent entre la masse et la broche POT... du 6561, et le boîtier VIC 6561 ne fait rien d'autre que mesurer la constante RC qui se retrouve dans le registre d'adresse 36872 ou 36873.



Si l'on met un condensateur C à l'extérieur du VIC entre les broches 7 et 8 du connecteur H, on pourra mesurer sa capacité ou mesurer une résistance R dans une autre gamme que les 200 k. Selon la résistance R, on aura différentes gammes de capacités mesurables.

Dans tous les cas, un étalonnage préalable est à faire.

On le voit, le VIC est capable d'une application d'instrumentation, intéressante au moins dans un contexte enseignement.

CRAYON LUMINEUX

Un crayon lumineux n'est rien d'autre qu'une cellule photoélectrique qui, pointée sur un point de l'écran, envoie une impulsion négative, lorsqu'elle décèle l'éclairage du point au moment précis où le balayage de l'écran atteint ce point. Or, le boîtier VIC 6561 "sait" à tout moment où il en est dans son balayage. Lorsqu'arrive l'impulsion du crayon lumineux, il mémorise les coordonnées du point en 36870(X) et 36871(Y).

Exemple d'utilisation

```
10 PRINT "Clr CHOISISSEZ"
20 PRINT "bbb Rvsc Sp Offc Sp Sp Sp
     1"
30  PRINT "bbbbbbbbbb  Rvsc  Sp Offc
     Sp Sp Sp 2"
40 FOR I = 1 TO 3000 : NEXT : REM 3
     SECONDES DE DELAI
50 X = PEEK(36870) : Y=PEEK(36871)
60 IF Y X 90 PRINT"VOUS AVEZ CHOISI
     1" : END
70 PRINT "VOUS AVEZ CHOISI 2"
```

· On indiquera dans le prochain numéro comment se bricoler un crayon lumineux. Mais le problème est qu'il faut une très grande précision optique pour que les registres captent des coordonnées précises. L'utilisation de toutes les possibilités du 6561 dans ce domaine est donc hors de portée d'un bricolage.

Daniel Jean DAVID

Correspondances VIC/CBM et CBM/VIC

La page suivante donne les correspondances des adresses CBM-VIC et VIC-CBM. Le tableau ci-dessous donne quelques adresses qui sont différentes selons les versions.

VIC	645	649	650	651	652
CBM8000	1020	227	228	229	
CBM4000G	1020	1003	1006	1001	1002

N.B. Les correspondances entre adresses concernant la gestion des cassettes sont données sous réserves.

CBM	VIC
0	0
.	.
2	2
3	7
.	.
4	8
5	11
.	.
13	19
17	20
.	.
140	143
141	160
.	.
143	162
144	788
.	.
149	793
150	144
151	203
152	653
155	145
.	.
157	147
158	198
159	199
160	148
161	200
163	201
164	202
165	149
166	197
167	204
.	.
170	207
172	208
173	151
.	.
181	159
183	163

CBM	VIC
185	164
.	.
190	168
193	169
195	171
196	209
198	211
199	172
.	.
204	177
205	212
206	180
.	.
212	186
213	213
214	178
215	179
216	214
217	215
218	187
219	188
220	216
221	189
223	191
224	217
.	.
248	241
249	192
251	193
252	194
256	256
.	.
511	511
512	512
.	.
591	600
593	601
.	.
632	640
634	828
.	.
825	1019

VIC	CBM
0	0
.	.
2	2
7	3
8	4
11	5
.	.
19	13
20	17
.	.
143	140
144	150
145	155
.	.
147	157
148	160
149	165
150	207
151	173
.	.
159	181
160	141
.	.
162	143
163	183
164	185
166	187
167	189
168	190
169	193
.	.
171	195
172	199
.	.
177	204
178	214
179	215
180	206
.	.
186	212
187	218
188	219

VIC	CBM
189	221
.	.
191	223
193	251
194	252
197	166
198	158
199	159
200	161
201	163
202	164
203	151
204	167
.	.
207	170
208	172
209	196
211	198
212	205
213	213
214	216
215	217
216	220
217	224
.	.
241	248
256	256
.	.
511	511
512	512
600	592
601	593
.	.
640	632
653	152
788	144
.	.
793	149
828	634
.	.
1019	825

Banc d'essai de la Cartouche VIC Programmer's Aid

L'EDEX du VIC est déja disponible !

Outre différentes cartouches de jeu et les extensions mémoire 3K, 8K.... trois cartouches essentielles dites "d'aide à la programmation" sont disponibles. Ce sont :

VIC1213 (294 F) moniteur langage machine
VIC1212 (294 F) "Programmer's Aid"
VIC1211 (411.60F) "Super Expander"

Nous parlerons du moniteur à l'occasion d'un "coin du débutant en langage machine" où nous verrons que cette cartouche offre toutes les fonctions indispensables pour introduire et mettre au point un programme en langage machine.
"Super Expander" offre des extensions à BASIC permettant de faire de la couleur, des graphiques haute résolution et de la musique plus commodément qu'avec les POKE fastidieux. Nous lui consacrerons un banc d'essai complet dans un prochain numéro. Programmer's Aid offre des commandes BASIC facilitant l'édition et la mise au point des programmes BASIC à la manière des extensions EDEX ou Toolkit sur CBM.

Sans vouloir préjuger des conclusions de ces bancs d'essai, nous pensons que ces trois cartouches sont à un titre ou à un autre, indispensables à l'utilisateur du VIC. Il se pose alors le problème de la connexion simultanée de ces cartouches, puisque le VIC n'a qu'un connecteur d'extension. La solution assez onéreuse est d'utiliser un coffret d'extension soit ARFON soit Commodore 1020 (coffrets permettant de loger le VIC, l'alimentation, le modulateur, sur lesquels on peut poser la télé, et qui ont un fond de panier à 6 ou 7 connecteurs). La société SEDERMI offre une solution moins luxueuse mais meilleur marché pour le même problème avec les cartes EXTVIC-BUS4. Les essais ont prouvé que cette carte à 340 Frs permettait d'avoir en fonctionnement parfait les trois cartouches "d'aide à la programmation" plus une carte mémoire 16 K. Cette carte permet la connexion de quatre cartouches simultanément et nous pensons que c'est suffisant car, avec les incompatibilités existant entre cartouches (p. ex. 2 cartouches de jeu s'excluent, Super Expander est incompatible avec la RAM 3K et avec les jeux, etc.) il sera bien rare d'avoir plus de 4 cartouches à connecter simultanément.

Les fonctions de Programmer's Aid

Les fonctions de Programmer's Aid se répartissent en 4 groupes :

1) Nouvelles commandes BASIC d'aide à l'écriture des programmes.

2) Nouvelles commandes BASIC d'aide à la mise au point des programmes.

3) Fonctions supplémentaires assignées à certaines touches pour abréger l'écriture.

4) Commandes de lancement, changement de mode, etc.

COMMANDES D'AIDE A L'ECRITURE

Ces commandes d'aide à l'écriture ou à l'édition des programmes sont très utiles et judicieusement choisies. Pour certaines notamment FIND et CHANGE, les syntaxes qui ont été choisies ne sont pas très pratiques. Passons les en revue.

AUTO départ,incrément : numérotation automatique.

CHANGE chaîne1,chaîne2,l1-l2: change toutes les occurrences de chaîne 1 en chaîne 2 dans la partie du programme comprise entre l1 et l2 (comme LIST).

Si chaîne1 et chaîne2 sont fournies entre guillemets, ce sont les occurrences qui font partie d'un texte entre guillemets qui seront changées.

DELETE l1-l2 : supprime les lignes du programme comprises entre l1 et l2 (syntaxe analogue à LIST, sauf que DELETE tout court est interdit)

FIND chaîne, l1-l2 : trouve toutes les occurrences de la chaîne dans l'intervalle des lignes l1-l2. Si chaîne est fournie entre guillemets ce sont les occurrences qui font partie d'un texte entre guillemets qui sont trouvées.

MERGE "nom de fichier".périph : charge le programme indiqué depuis le périphérique spécifié sans effacer le programme déjà en mémoire. Il y a fusion des deux programmes : si le fichier contient une instruction de même numéro qu'une instruction déjà présente, c'est celle du fichier qui prévaudra.

Cette commande justifie à elle seule l'achat de la cartouche Programmer's Aid, à notre avis. Elle fonctionne aussi bien sur disque que sur cassette.

RENUMBER départ,incrément : renumérote tout le programme. Il y a bien sûr correction des étiquettes des GOTO, etc...

COMMANDES D'AIDE A LA MISE AU POINT

Les principales commandes de cette catégorie sont TRACE qui fait afficher les numéros d'instructions à mesure qu'on passe dessus et STEP qui permet l'exécution en pas-à-pas. Ces commandes sont d'un intérêt fondamental. Voici les autres, pour être complet.

DUMP : fait afficher toutes les variables et leurs valeurs, lors d'un arrêt. Très utile pour comprendre pourquoi un programme ne marche pas. Comme dans tous les systèmes analogues, les tableaux ne sont pas listés, ce qui est dommage.

HELP : Après une erreur, visualise l'endroit ou l'erreur s'est produite dans l'instruction.

OFF : sortie des modes TRACE et STEP.

STEP : passe en mode pas-à-pas où une instruction est exécutée à chaque fois qu'on appuie sur SHIFT.

TRACE : passe en mode TRACE où les numéros d'instruction sont affichés dans le coin supérieur droit de l'écran à mesure qu'on les exécute. Attention, cela trouble l'affichage dans cette zone.

FONCTIONS SUPPLEMENTAIRES DES TOUCHES :

Programmer's Aid permet d'assigner aux 4 touches de fonction à droite du clavier la chaîne de caractères que vous voulez, ce qui peut faire gagner énormément de temps de frappe. C'est l'objet de la Commande : KEY n,"texte" n est le numéro de la touche conformément au tableau ci-dessous. Le retour chariot peut même être inclus par :

```
KEY n,"texte" + CHR$(13)
```

Si vous n'utilisez pas KEY, des fonctions sont préassignées d'office. Il y a deux jeux de préassignations qui correspondent à deux modes de fonctionnement appelés PROG (programme) et EDIT (édition). Les commande EDIT et PROG permettent de changer de mode.

Voici les préassignations :

touches		mode PROG	mode EDIT
f1/f2	(1)	LIST	LIST
f1/f2+SHIFT	(5)	MID$(	AUTO
f1/f2+CTRL	(9)	EDIT C/R	PROG C/R
f3/f4	(2)	RUN C/R	RUN C/R
f3/f4+SHIFT	(6)	LEFT$(	DELETE
f3/f4+CTRL	(10)	GOSUB	RENUMBER
f5/f6	(3)	GOTO	FIND
f5/f6+SHIFT	(7)	RIGHT$(	CHANGE
f5/f6+CTRL	(11)	RETURN	MERGE
f7/f8	(4)	INPUT	TRACE C/R
f7/f8+SHIFT	(8)	CHR$(	STEP C/R
f7/f8+CTRL	(12)	STR$(	OFF C/R

( ) numéro de la touche pour la commande KEY
C/R Le retour chariot fait partie de la fonction

Enfin, Programmer's Aid ajoute des fonctions obtenues avec la touche CTRL. Ce sont :

CTRL A défilement du listing vers le haut

CTRL E sortie du mode guillemets en cours d'insertion

CTRL L efface tous les caractères de la ligne, après le curseur

CTRL N efface tous les caractères de l'écran, après le curseur

CTRL Q défilement du listing vers le bas, avec réapparition des lignes qui étaient sorties par le haut de l'écran.

CTRL U effacement de la ligne où se trouve le curseur.

C'est CTRL Q qui est le plus intéressant et il compense presque la limitation de la taille de l'écran.

COMMANDES DE LANCEMENT :

Le lancement de Programmer's Aid s'effectue par SYS 28681, ou plus facile à se rappeler :
SYS 7 * 4096 + 9

On est alors en mode PROG. On a vu qu'on changeait de mode par EDIT avec retour par PROG.

La commande KILL désactive Programmer's Aid. En effet, certaines commandes font perdre du temps à l'interpréteur BASIC. Une fois le programme au point, on a intérêt à les désactiver. Il faut noter que KILL ne désactive pas les touches de fonction, c'est un bon point.

Conclusions

Programmer's Aid est, à notre avis, une cartouche indispensable à tous ceux qui souhaitent développer des programmes un peu conséquents. Elle n'est pas d'un prix trop élevé. Sa présence porte le BASIC du VIC à un niveau excellent, sans concurrence sur le marché à son niveau de prix.

Notons pour terminer que Programmer's Aid est livrée avec une notice en français complète et bien présentée.

Pierre Etienne THALBERG

Gestion de Comptes

Programme minimum de gestion de compte sur VIC20 ou, en modifiant les adresses relatives à la mémoire d'écran, sur CBM. Volontairement écrit pour "tourner" sur la configuration minimale du VIC, afin de pouvoir être mis en oeuvre par tous, il nécessitera un certain effort de compréhension de la part du débutant.

Introduction

Le VIC permet normalement d'afficher 23 lignes de 22 caractères.

Une gestion de fichier sur cassette n'est facile que si le fichier géré tient entièrement en mémoire : toutes les opérations sur le fichier sont faites en mémoire, puis, en fin de traitement, le fichier est intégralement recopié sur cassette. Compte tenu des limitations de l'affichage du VIC et de la taille du fichier qui doit tenir en mémoïre, un certain nombre de choix ont été faits :

- les informations enregistrées sur chaque opération sont :

. une date, limitée à 8 caractères

. le type de l'opération, limité à 10 caractères. Aucun contrôle n'est fait et l'utilisateur peut mettre ce qu'il veut, par exemple "chèque", "virement"...

. un libellé ou "raison du paiement", limité à 10 caractères. Aucun contrôle n'est fait et l'utilisateur peut mettre ce qu'il veut, par exemple "facture EDF", "paie"....

. un montant limité à 99 999.99 et tronqué à 2 décimales.

. le sens : "débit" ou "crédit" est traduit par le signe du montant.

- l'écran est partagé en 3 parties fonctionnelles :

. La partie supérieure où sont affichés :

+ le nom du programme et 3 nombres indiquant respectivement le numéro d'ordre dans le fichier de l'opération affichée en tête de la partie centrale de l'écran, le nombre d'opérations enregistrées dans le fichier et le nombre maximum d'opérations prévues dans le fichier.

+ Le nom du compte correspondant au fichier.

+ Le solde du compte avec l'indication C ou D selon qu'il est Créditeur ou Débiteur.

. La partie centrale où sont affichés le type d'opération (limité aux 3 premiers caractères), la date, le montant de 6 opérations. En outre la ligne représentant une opération se terminera par un carré rouge si à l'issue de cette opération le compte est à découvert.

Cette partie se terminera par l'affichage du solde à l'issue de la dernière opération affichée.

. La partie inférieure de l'écran réservée au dialogue entre le programme et l'utilisateur.

- Les fonctions du programme sont :

. prise en compte d'un débit ou d'un crédit

. édition (temporisée) des

mouvements enregistrés, avec possibilité d'arrêt de l'édition, de reprise ou de sortir de la fonction d'édition.

. modification d'un mouvement.

- 9 sous-programmes ont été écrits pour réaliser 9 fonctions particulières. Ils ont été regroupés à la fin du programme et ont des numéros de ligne au moins égaux à 1000.

Sous-programmes

On peut tout d'abord rappeler les 2 raisons d'écrire un sous-programme :

1 - Ecrire une seule fois un ensemble d'instructions réalisant une fonction requise à plusieurs endroits du programme.

2 - Alléger le "corps du programme" afin d'en faciliter la compréhension. Bien souvent les 2 raisons jouent simultanément.

```
1000 PRINT"PLACEZ LA CAS
  SETTE-"R$J$", PUIS TAPEZ C"
1010 GETX$:IFX$<>"C"GOTO1010
1020 RETURN
```

```
5000 IFY$="C"THENPRINT"CREDIT"
5010 IFY$="D"THENPRINT"DEBIT"
5020 RETURN
```

Ces 2 sous-programmes ne présentent pas de difficulté. On peut cependant remarquer l'utilisation presque systématique qui a été faite des variables alphanumériques R$, J$ ainsi que d'autres, afin d'économiser la mémoire : J$ prend moins de place que "FICHIER", R$ prend moins de place que CHR$(13) afin de représenter le caractère "Return".

```
9500 POKE781,I:POKE782,0
     :SYS16*4095:RETURN
```

Ce sous-programme a déjà été indiqué page 51 dans le VIC à BRAC du numéro 4 de La Commode. Il correspond ici à un cas particulier: placer le curseur au début de la ligne I+1 de l'écran.

```
9800 J1=DA+U*J:FORK=J1TOJ1+21
     :POKEK,M:POKEK+DC,C
     :NEXTK:RETURN
```

DA vaut 7680, c'est-à-dire l'adresse du début de la table des codes des caractères de l'écran.

DC vaut 30720, c'est-à-dire la valeur à ajouter à l'adresse contenant le code caractère d'une position (lignes J+1, colonne K-J1+I par exemple) de l'écran pour avoir l'adresse contenant le code. POKE K, M place le code caractère M dans la table des codes des caractères de l'écran et POKE K+DC,C place le code couleur C dans la position correspondante de la table des codes des couleurs de l'écran.

Ce sous-programme ligne 9800 place donc le caractère de code M, de couleur C sur toute la ligne J+1 de l'écran. Il est utilisé à 2 fins dans le programme : mettre certaines lignes à blanc avant d'y réécrire, et tracer les traits séparant l'écran en 3 parties.

```
9600 GOSUB9500:M=32
     :FORJ=ITO21:GOSUB9800
     :NEXT:GOSUB9500:RETURN
```

32 est le code de l'espace. Ce sous-programme place le curseur sur la ligne I+I de l'écran en "nettoyant" les lignes suivantes .

```
9900 Y$=LEFT$(RIGHT$(T$+T$
     +STR$(Y+.01*((Y<0)
     +.5)),M+1),M)
9910 IFABS(Y)<.005THEN
     Y$=RIGHT$(T$+T$+"0.00",M)
9920 PRINTY$;:RETURN
```

C'est ce sous-programme qui permet d'éditer les montants ou les soldes en colonnes bien alignées. Il convient pour des nombres signés d'au plus 2 décimales. M donne la largeur totale de la zone d'édition; T$ + T$ impose la présence d'au moins 10 caractères à gauche du point décimal, ainsi M peut valoir jusqu'à 13 (10 caractères + le point décimal + 2 décimales). Au

besoin, pour être sûr que Y n'a que 2 décimales, avant l'appel du sous-programme il peut être utile de faire Y = INT(Y*100 + .5) / 100.

```
9000 PRINT"SOLDE :";
  :Y=ABS(S(I)):M=13:GOSUB9900
9010 PRINT" "CHR$(67
  -(S(I)<0)):C=3:M=67
  :GOSUB9800:RETURN
```

Ce sous-programme édite le solde S(I), après le (I+1)ème mouvement. Pour I=N, N+I étant le nombre de mouvements, cela édite le solde actuel du compte. S(I)(0 vaut -1 ou 0 selon que S(I) est négatif ou non ; 67 et 68 sont les codes ASCII des caractères C et D.

```
7000 S(0)=S0+M(0):IFN>0THEN
   FORI=1TON:S(I)=INT((S(I-1)
     +M(I))*100+.5)/100:NEXT
7010 P=P+1:RETURN
```

Il s'agit de calculer ou de recalculer le solde à la suite de chacun des mouvements. Ce sous-programme est appelé à la suite de l'enregistrement d'un nouveau mouvement ou d'une modification d'un mouvement plus ancien. L'instruction P=P+1 (P=1 suffirait) permet alors, en fin de programme, de savoir si le fichier a été modifié ou non, donc s'il doit être réécrit ou non.

Le sous-programme ci-dessous, à chaque appel, refait complètement l'affichage des 2 premières parties de l'écran. Les opérations affichées dans la 2ème parties sont celles comprises entre la (N1+1)ème et la (N2+1)ème comprises. La ligne 6040 permet d'afficher un carré rouge à la fin de la ligne "opération" lorsqu'à la suite de celle-ci le compte est à découvert. POKE J,M place dans la table des codes soit un espace (=32) soit un espace inversé (=32 + 128) selon que le solde est positif ou négatif.
POKE J+DC,2 place la couleur rouge dans la position correspondante de la table des couleurs ; si M vaut 32 cela n'a pas d'effet.

Corps du programme

Du début à la ligne 10, les constantes numériques et alphanumériques sont affectées de leur valeur ou dimensionnées.

Les lignes 20 et 30 correspondent au dialogue de début du programme : selon que le fichier existe déjà ou non il faut d'abord le lire ou au contraire acquérir son nom et le solde initial.

Les lignes 50 et 70 font ce début de création de fichier tandis que les lignes 120 à 160 correspondent à la lecture du fichier.

Les lignes 170 à 200 affichent la liste des fonctions possibles : l'utilisateur doit alors taper la première lettre de la fonction désirée ; sa réponse est lue par GET.

La fonction F ou "fin" est réalisee des lignes 220 à 280 et consiste essentiellement à réécrire le fichier (à jour) sur cassette.

La fonction E ou "édition" est réalisée des lignes 310 à 340. L'affichage de l'ensemble des opérations du fichier est fait par affichages successifs de 6 opérations. Chaque nouvel affichage reprend les 5 dernières opérations de l'affichage précédent et en

```
6000 PRINTA$"■"SPC(21-LEN(STR$(N1)+STR$(N)+STR$(L)))N1"/"N"/"L:PR
                                                          INTJ$IN$
6005 J=4:I=N:GOSUB9000:PRINT"DES. "C$T$D$
6010 FORI=N1TON2
6020 PRINT"◣"LEFT$(D$(I)+T$,3)"▣"LEFT$(DA$(I)+T$+T$,8);
6030 Y=M(I):M=9:GOSUB9900:PRINT"":M=32:IFS(I)<0THENM=160
6040 J=DA+(I-N1+6)*U+21:POKEJ,M:POKEJ+DC,2
6050 NEXT:I=12:GOSUB9500:I=N2:J=13:GOSUB9000:RETURN
```

ajoute une. Entre 2 affichages successifs la boucle FOR J=0 TO 200... NEXT J des lignes 320 à 340 a un double rôle :

- assurer une temporisation entre 2 lectures, peut-être un peu longue : il suffit de mettre une valeur inférieure à 200 pour l'écourter.

- permettre en appuyant sur la touche A, de s'arrêter sur un affichage.

Il y a alors 2 façons de repartir :

- en appuyant sur la touche D, les affichages successifs reprennent.

- en appuyant sur la touche F on sort de la fonction d'édition pour sélectionner une nouvelle fonction.

Les fonctions "C" et "D" c'est à dire "crédit" et "débit" sont trop semblables pour être traitées séparément. Elles sont réalisées des lignes 360 à 420. et permettent de rentrer successivement :

- le mode de paiement. "XXX" par défaut c'est-à-dire si la touche RETURN est appuyée tout de suite La fin de la ligne 370 :
I=15 : GOSUB 9600 sert à replacer le curseur et à nettoyer le bas de l'écran En effet une lecture par INPUT ne permet pas un contrôle permanent et strict de l'affichage.

- la date, avec le même nettoyage systématique de l'écran, suivi du réaffichage (bien placé) de la date rentrée. Si aucune date n'est rentrée, la date prise en compte est celle de l'opération précédente.

- La raison de l'opération. Par défaut c'est la valeur "chaîne vide" qui est prise.

- le montant. Par défaut c'est une valeur nulle qui est acquise avec la possibilité de modifier ultérieurement tout ou partie de l'opération. La valeur lue est limitée à 2 décimales et est refusée si elle est supérieure à 99 999,99.

Sur la ligne 420 l'expression:
2*(Y$="D")+I,
vaut -1 ou +1 selon que Y$ vaut "D" ou non, et permet de coder le sens "Débit" / "Çrédit" par le signe du montant.

- La fonction M ou modification des lignes 505 à 580 diffère peu des fonctions C et D si ce n'est par un dialogue légèrement plus riche précisant pour chaque valeur lue la valeur ancienne. C'est bien sûr l'ancienne valeur qui est reprise par défaut.

Adaptations

VIC 20 de base ou + 3 K

DA = 7680
DC = 30720
U = 22

VIC 20 >= 8K

DA = 4096
DC = 33792
U = 22

PET CBM3000/4000(40 COL.)

DA = 32768
DC : il faut supprimer toutes les instructions avec DC.
U = 40

CBM8000

DA = 32768
DC : il faut supprimer toutes les instructions avec DC.
U = 80

Dans le sous-programme 9600, la valeur maximale 21 pour la variable J correspond au nombre de lignes sur l'écran. Pour un écran de 25 lignes, il vaut mieux mettre 23 au lieu de 21.

Le sous-programme 9500 est aussi à refaire pour CBM :

```
9500 POKE 216,I : POKE 198,0 :
SYS 57949 : RETURN
```

De plus:

- en 6000 ... SPC(21..., 21 doit être remplacé par 39 pour un écran 40 colonnes ou par 79 pour un écran 80 colonnes (soit U-1).

- en 9000, juste avant ":C=3", ajouter : "PRINT", du fait que seulement sur le VIC 20 l'impression du C ou du D se fait en fin de ligne provoquant ainsi un saut de ligne supplémentaire.

Avec cet ensemble de modifications, le programme sera adapté aux différents PET/CBM. Cependant l'écran de ces modèles sera sous-exploité puisque l'affichage se fera sur 23 lignes de 22 colonnes mais cela marchera...

Sur le VIC 20 de base sans aucune extension, on peut envisager de 10 à 30 opérations au maximum sur le fichier selon que les valeurs données aux D$(I) et EC$(I) sont de 10 caractères au moins en moyenne (jusqu'à 3).

Avec, ne serait-ce que l'extension de 3 K, on peut envisager des fichiers plus importants, de l'ordre de 200 opérations pour un fichier ; pour la gestion d'un compte personnel cela correspondra généralement à plusieurs mois.

Table des variables

DA : adresse du début de la table des codes des caractères affichés.

DC : valeur à ajouter à DA pour avoir l'adresse du début de la table des codes de couleur

D$(L) : tableau indicé de 0 à L contenant les "modes de paiement" des N+1 opérations enregistrées.

DA$(L) : Tableau indicé de 0 à L contenant les "dates" de paiement des N+1 opérations enregistrées.

EC$(L) : Tableau indicé de 0 à L contenant "raisons du paiement" des N+1 opérations enregistrées.

I : variable de travail

IN$: contient le nom du fichier

J : variable de travail.

K : variable de travail.

L : dimension des tableaux, limitant à L+I le nombre maximal d'opérations possibles

M(L) : tableau de L+I éléments, contenant les "montants" des N+I opérations enregistrées.

N : N+1 = le nombre d'opérations enregistrées.

N1,N2,N3 : pointeurs sur les opérations éditées dans la 2ème partie de l'écran.

P : variable d'état, non nulle si le "fichier" a été modifié (fonctions C,D ou M).

R$: = CHR$(13) : retour chariot

S : = 99 999.99 montant maximum d'un crédit/débit (choix découlant des contraintes sur l'affichage).

S(L): L+1 éléments possibles : soldes à l'issue des N+1 opérations effectives.

S0 : solde initial

U : nombre de colonnes de l'écran

X : variable de travail (entrées numériques).

X$: variable de travail (entrées alphanumériques).

Y : variable de travail

Y$: variable de travail (entrées alphanumériques et sous-programme 9900.

Z : pointeur de l'opération modifiée.

Les variables T$, S$, A$, B$, C$, D$, E$, F$, G$, H$, I$ et J$ sont des constantes initialisées au début du programme, lignes 2 à 8.

Daniel TRECOURT

```
1 DEFFNA(X)=INT(X*100+.5)/100
2 L=10:R$=CHR$(13):T$="          ":S$=R$+T$:U=22:DA=7680:DC=30720
4 A$="■■GESTION DE COMPTE■":B$="MODE DE":C$="DATE":D$="MONTANT"
6 E$="TROP ELEVE: IGNORE":F$="■C■REDIT":I$="■D■EBIT?":G$="RAISON DU"
8 H$=" PAIEMENT":J$="FICHIER ":S=99999.99
10 DIMD$(L),DA$(L),EC$(L),M(L),S(L)
20 PRINTA$R$R$"CREATION?(■O■UI/■N■ON)?"
30 GETX$:IFX$=""GOTO30
40 IFX$<>"O"GOTO100
50 PRINT"■■CREATION DE "J$R$"■NOM DU "J$:INPUTIN$:IN$=LEFT$(IN$,13)
70 PRINT"SOLDE INITIAL":INPUTS0:S0=FNA(S0):N=-1:GOTO170
100 IFX$<>"N"GOTO30
120 GOSUB1000:OPEN1,1,0:INPUT#1,IN$,N,S0
140 FORI=0TON:INPUT#1,D$(I),DA$(I),EC$(I),M(I),S(I):NEXT:CLOSE1
150 N1=N-5:IFN1<0THENN1=0
160 N2=N:GOSUB6000
170 PRINT"QUELLE FONCTION?":IFN=-1GOTO190
180 IFN=LTHENPRINTR$J$"PLEIN:"R$"AUCUN DEBIT OU CREDIT"
185 PRINTS$"■E■DITION"S$"■F■IN"S$"■M■ODIFICATION";
190 IFN<L THENPRINTS$F$S$I$
200 GETY$:IFY$=""GOTO200
210 IFY$<>"F"GOTO300
220 IFP=0THENPRINTJ$"NON MODIFIE":GOTO280
230 PRINT"ECRITURE DU "J$:GOSUB1000:OPEN1,1,1,IN$:PRINT#1,IN$R$N;R$S0
260 FORI=0TON:PRINT#1,D$(I)R$DA$(I)R$EC$(I)R$M(I)R$S(I)R$:NEXT:PRINT#1:CLOSE1
280 PRINT"FIN DU PROGRAMME":END
300 IFY$<>"E"GOTO350
310 N1=0:N3=5:IFN3>NTHENN3=N
320 FORN2=N3TON:GOSUB6000:N1=N1+1:FORJ=0TO200:GETY$:IFY$<>"A"ORN2=NGOTO340
330 J=200:GETY$:IFY$<>"D"ANDY$<>"F"GOTO330
335 IFY$="F"THENN2=N
340 NEXTJ:NEXTN2:GOTO170
350 IFY$<>"C"ANDY$<>"D"ORN=LGOTO500
360 I=14:GOSUB9600:GOSUB5000
370 N=N+1:PRINTB$H$:X$="XXX":INPUTX$:D$(N)=LEFT$(X$,10):I=15:GOSUB9600
380 PRINTD$(N):PRINTC$:X$="":IFN>0THENX$=DA$(N-1)
385 INPUTX$:DA$(N)=LEFT$(X$,8):I=16:GOSUB9600:PRINTDA$(N)
390 PRINTG$H$:X$="":INPUTX$:EC$(N)=LEFT$(X$,10):I=17:GOSUB9600:PRINTEC$(N)
410 PRINTD$:X=0:INPUTX:X=FNA(ABS(X))
415 IFX>STHENI=18:GOSUB9600:PRINTE$:GOTO410
420 M(N)=X*(2*(Y$="D")+1):GOSUB7000:GOTO150
500 IFY$<>"M"GOTO200
505 I=14:GOSUB9600:PRINT"MODIFICATION":PRINT"NO.DU MOUVT. A MODIF.":Z=-1
510 INPUTZ:IFZ<0ORZ>NGOTO510
520 I=15:GOSUB9600:PRINT"MOUVT. NO."Z:Y$=CHR$(67-(M(Z)<0)):PRINTF$" OU "I$
525 GETY$:IFY$<>"C"ANDY$<>"D"GOTO525
527 I=16:GOSUB9600:GOSUB5000
530 PRINTB$H$R$D$(Z):INPUTD$(Z):D$(Z)=LEFT$(D$(Z),10):I=17:GOSUB9600:PRINTD$(Z)
540 X$=DA$(Z):PRINTC$X$:INPUTX$:DA$(Z)=LEFT$(X$,8):I=18:GOSUB9600:PRINTDA$(Z)
550 X$=EC$(Z):PRINTG$H$R$X$:INPUTX$:EC$(Z)=LEFT$(X$,10)
555 I=19:GOSUB9600:PRINTEC$(Z)
560 X=M(Z):PRINTD$ABS(X):INPUTX:X=FNA(ABS(X))
570 IFX>STHENI=20:GOSUB9600:PRINTE$:GOTO560
580 M(Z)=X*(2*(Y$="D")+1):GOSUB7000:GOTO150
1000 PRINT"■PLACEZ LA CASSETTE-"R$J$", PUIS TAPEZ C"
1010 GETX$:IFX$<>"C"GOTO1010
1020 RETURN
5000 IFY$="C"THENPRINT"CREDIT"
5010 IFY$="D"THENPRINT"DEBIT"
5020 RETURN
6000 PRINTA$"■"SPC(21-LEN(STR$(N1)+STR$(N)+STR$(L)))N1"/"N"/"L:PRINTJ$IN$
6005 J=4:I=N:GOSUB9000:PRINT"DES. "C$T$D$
6010 FORI=N1TON2
6020 PRINT"■"LEFT$(D$(I)+T$,3)"■"LEFT$(DA$(I)+T$+T$,8);
6030 Y=M(I):M=9:GOSUB9900:PRINT"":M=32:IFS(I)<0THENM=160
6040 J=DA+(I-N1+6)*U+21:POKEJ,M:POKEJ+DC,2
6050 NEXT:I=12:GOSUB9500:I=N2:J=13:GOSUB9000:RETURN
7000 S(0)=S0+M(0):IFN>0THENFORI=1TON:S(I)=INT((S(I-1)+M(I))*100+.5)/100:NEXT
7010 P=P+1:RETURN
9000 PRINT"SOLDE :";:Y=ABS(S(I)):M=13:GOSUB9900
9010 PRINT" "CHR$(67-(S(I)<0)):C=3:M=67:GOSUB9800:RETURN
9500 POKE781,I:POKE782,0:SYS16*4095:RETURN
9600 GOSUB9500:M=32:FORJ=ITO21:GOSUB9800:NEXT:GOSUB9500:RETURN
9800 J1=DA+U*J:FORK=J1TOJ1+21:POKEK,M:POKEK+DC,C:NEXTK:RETURN
9900 Y$=LEFT$(RIGHT$(T$+T$+STR$(Y+.01*((Y<0)+.5)),M+1),M)
9910 IFABS(Y)<.005THENY$=RIGHT$(T$+T$+"0.00",M)
9920 PRINTY$;:RETURN
```

VIC à BRAC

Méthode pour sauver des octets langage machine contenus entre A et B

Si : A = LA + 256 * HA ; B = LB + 256 * HB

```
POKE 186,1 : POKE 780,0 : POKE
781,LB + 1 : POKE 782,HB
POKE 0,LA : POKE 1,HA : SYS 63093
```

que l'on rechargera par LOAD"",1,1. Le programme résidant ne sera pas écrasé à condition de restaurer les pointeurs (45,46) et (47,48).

Utilisation mode multicouleur du VIC

Un petit programme assez spectaculaire pour ceux qui ont un moniteur couleur :

```
  5 REM VIC ARTISTE
 10 DEF FNR(X) = INT(RND(1)*X)
 20 PRINT "CLR" :G=6144:CO=30720
 30 FORI = 7680 TO 8185 : POKE I,0:
    POKE I+CO, FNR(8) + 8 :
    NEXT : POKE 36869,254
 40 FOR I = 1 TO 3
 50 POKE 36878, FNR(16) * 16
 60 FOR J = G TO G + 7 : POKE J,A
 70 FOR T = 1 TO110 : NEXT : NEXT J
 80 FOR T = 1 TO 300 : NEXT
 90 FOR J = G + 7 TO G STEP - 1 :
    POKE J, FNR(256)
100FOR T = 1 TO 110 : NEXT : NEXT J
110A = FNR(256) : NEXT I
120POKE 36879, FNR(256) : GOTO 40
```

Vérifier si une cartouche d'extension RAM est bien connectée.

Il arrive fréquemment que l'on doive changer de cartouches, par exemple pour faire jaser les amis de passage. A la longue, les connexions deviennent plus lâches et même si tout semble normal à l'écran, des initialisations ne sont pas faites. Par exemple, avec une extension RAM, les programmes ne chargeront pas et on accusera le magnétophone ou bien des messages d'erreurs farfelus apparaissent lors de l'exécution. Je conseille alors, juste après avoir inséré la cartouche, de faire SYS 64802 (initialisation). Si l'affichage devient fantaisiste, la cartouche est mal connectée.

Programmer la couleur des PRINT dans un programme

POKE 646, code couleur.

Gagner de la place lors de la construction des caractères graphiques.

La définition des caractères graphiques demande beaucoup de mémoire pour stocker les nombres (dans des DATA par exemple) et celle-ci est précieuse dans un VIC de base. Une première solution a été proposée par Daniel Jean DAVID dans la "Découverte du VIC" pour la carte de France. Voici la mienne :

On introduit : 0 GOTO 10
1 REM @@@....
2 REM @@@....
.
.
10 (le programme)

En principe l'éditeur n'admet que 83 @ par REM, mais en modifiant les liens entre les lignes, on peut aller jusqu'à 249 @

Le premier @ est à l'adresse 4110. Par des POKE, on peut remplacer les @ par les octets définissant les caractères. Il faut bien sûr éviter l'octet 0 (fin de ligne) mais dans presque tous les cas un nombre entre 1 et 255 n'est pas utilisé, on remplace 0 par ce nombre. Lors de la translation dans le nouveau générateur graphique, on fera un test sur ce nombre en le remplaçant par 0.

On évitera de lister les REM à cause des résultats bizarres par LIST 10-.On peut même obtenir des "Syntax error" d'où la présence du GOTO 10 au début du programme.

Lecture du joystick

On recherche en priorité la rapidité. Voici la méthode que j'utilise en attendant une meilleure :

```
 5 Q1 = 37151 : Q2 = 37152 :
   POKE 37154,127 : P = 8000 :
   POKE 36879,8 : PRINT "CLR" :
   CO = 30720

10   T1 = PEEK(Q1):T2 = PEEK(Q2):
     D0 = SGN(T1 AND 4) :
     D1 = SGN(T1 AND 8) :
     D2 = SGN(T1 AND 16)

20   FB = T1 AND 32 :
     D3 = SGN(T2 AND 128)
     P = P-22*(D1-D0)+ D2-D3 :
     POKE P,0
     IF FB GOTO suite
          (FB = 0, le bouton rouge
          a été pressé (on efface)
          POKE P + CO,0

suite GOTO 10
```

Hervé LE MARCHAND

PRECISION

Il ne s'agit pas d'un erratum mais presque ; les deux points ayant mal marqué certains exemplaires de la Commode numéro 3 page 39. L'instruction 320 de MASTERMIND VIC doit se lire :

```
320    FORI=0TOV:PRINT"damier     2
espaces";:B(I)=-1:NEXT:PRINT"2
curseurs gauche";
```

ERRATUM

Dans les registres d'entrée-sortie du VIC (La Commode numéro 4 page 37), l'entrée JOY3 et la sortie écriture cassette ont été attribuées au port B du mauvais 6522. C'est au port B du 6522 numéro 2 qu'il faut les attribuer, respectivement au bit 7 et au bit 3, adresse 37152.

Ainsi pour déceler si JOY3 est actionné, il faut faire :

```
IF PEEK(37152) AND 2↑7 THEN...
```

Aussi les entrées JOY0, JOY1, JOY2 et crayon lumineux sont retournées, dans le port A, adresse 37151 : elles sont respectivement aux bits 2, 3, 4 et 5 et non 5, 4, 3, 2 comme on l'avait indiqué.

Ainsi pour déceler une impulsion du crayon lumineux, il faut faire :

```
IF PEEK(37151) AND 2↑5 THEN...
```

PRECISION

PROCEP nous communique que EDEX 3.3 (c'est-à-dire pour 4000) marche à la fois pour le petit et le grand écran. Prix 529,20 F TTC

sur CBM 8096

MASTER 2

la puissance en plus

COMPLET

MASTER 2 est le seul progiciel d'aide au développement intégrant des outils complets dans tous les domaines. Ces outils se commandent par de nouvelles instructions BASIC, facilitant ainsi assimilation et maintenance. MASTER 2 c'est une gestion de fichier MULTICLES et MULTICRITERES, une gestion d'ECRAN pour des applications "PRO", une gestion des EDITIONS par masque, un TRI multicritères, une précision de calcul sur 22 CHIFFRES, un contrôle automatique de validation des dates... et un système de protection de vos applications contre le listing et les duplications grâce à une CLE ELECTRONIQUE.

RAPIDE

Les programmes utilitaires GENECRAN, GENEDIT et GENEFILE vous permettent de développer et mettre au point vos applications rapidement, comme avec un logiciel "OUVERT", en mode direct. Ils sont livrés avec MASTER 2.

SOUTENU

La documentation de MASTER 2 est complète et en FRANÇAIS. Des exemples illustrent chaque fonction et de nombreux programmes de démonstration sont fournis. Le soutien c'est aussi des journées de formation organisées régulièrement.

PERFORMANT

Avec MASTER vous programmez vite et facilement. Vous disposez enfin des outils désirés de tous les développeurs. MASTER est entièrement écrit en langage machine 6502 et est intégré au BASIC. C'est pour vous la garantie d'un produit performant.

SÛR

MASTER est conçu pour les professionnels. Il est développé par la S.S.C.I. française MICRO APPLICATION. Présenté au SICOB 81 près de 2000 utilisateurs l'ont déjà choisi. Il est maintenant distribué en Europe et aux Etats-Unis. C'est pour vous la garantie d'un produit sûr.

RENTABLE

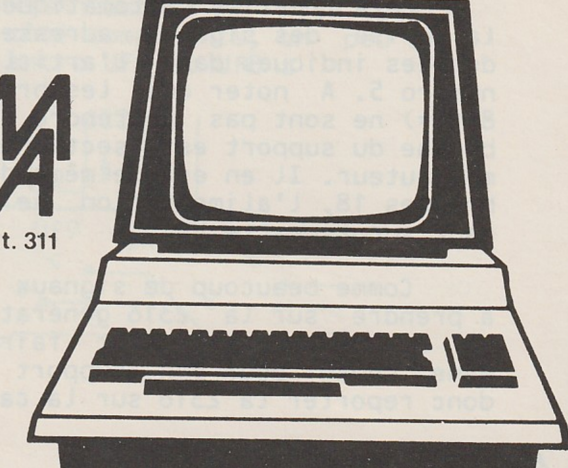
La clé électronique de MASTER permet de protéger vos applications et ainsi votre investissement. L'ensemble MASTER-CBM permet la réalisation simple d'applications de gestion de haut niveau. C'est pour vous, rapidement, la garantie d'un produit rentable.

MICRO APPLICATION continuera à développer MASTER et à l'implémenter sur les nouveaux systèmes COMMODORE avec le souci constant de la compatibilité ascendante.

MASTER existe en version de développement et en version d'exploitation

commodore

MA



MASTER est développé par MICRO-APPLICATION.

73, rue de Turbigo 75003 Tél. : (1) 274.70.55 Télex : 270105 F TXFRA/ext. 311

Il est distribué par :



19-21, RUE MATHURIN-RÉGNIER - 75015 PARIS
TÉL. : 306.82.02 - TÉLEX : 204 875 F.

Affichage Haute Résolution (suite)

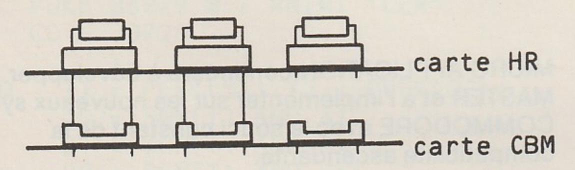
Nous voici maintenant au pied du mur pour réaliser effectivement notre carte haute résolution. Les points les plus délicats sont les modifications à faire sur la carte principale du CBM et la prise des signaux voulus.

Cet article aborde aussi les principes du logiciel à développer.

REALISATION DE LA CARTE

La carte haute résolution se réalise suivant le principe de la figure 3 de l'article du numéro 5. On donne aux deux 2114 du bloc 0 exactement la même disposition que sur la carte du CBM et on emploiera pour elles des supports à wrapper alors que tous les autres circuits ont des supports à souder.

La carte HR sera alors disposée au dessus de la carte du CBM, les queues plus longues des supports à wrapper venant s'enficher dans les supports des 2114 de la carte CBM :



Cela réalise automatiquement la prise des signaux adresses et données indiqués dans l'article du numéro 5. A noter que les broches 8(CS*) ne sont pas à prendre : la broche du support est sectionnée à mi-hauteur. Il en est de même des broches 18, l'alimentation se faisant sur la carte HR.

Comme beaucoup de signaux sont à prendre sur la 2316 génératrice de caractères, on peut faire la même chose pour son support et donc reporter la 2316 sur la carte HR. Nous ne l'avons pas fait pour notre part, car l'enfichage commencerait à devenir difficile (60 pattes à enficher simultanément).

Les seize 2114 sont à relier broches pour broches en parallèle sauf que, pour les lignes de données (br 11-14), on forme deux groupes de 8 et pour les CS*(br 8), on forme 8 groupes de 2. Les CS* iront aux 8 sorties du 74LS138, br 7 et 9-15, les CS* des 2114 qui remplacent celles de la carte CBM allant à la sortie 0 (br 15).

On a mis un condensateur de découplage (de 10 nF à 0,1 μF) entre le 0V et le +5V, toutes les deux 2114. Leur nécessité n'est pas certaine.

Les seuls autres circuits intégrés de la carte sont :

le décodeur 74LS138, le sélecteur 74LS157, le latch 74LS373 et le nand 74LS00. Leur connexion suit le schéma de l'article du numéro 5 : les données mémoire 2114 vont en entrée du 74LS373 (br impaires), tandis que les sorties (br paires) vont aux lignes de données du générateur 2316 (appelées motif dans la figure 3 page 31 du numéro 5).

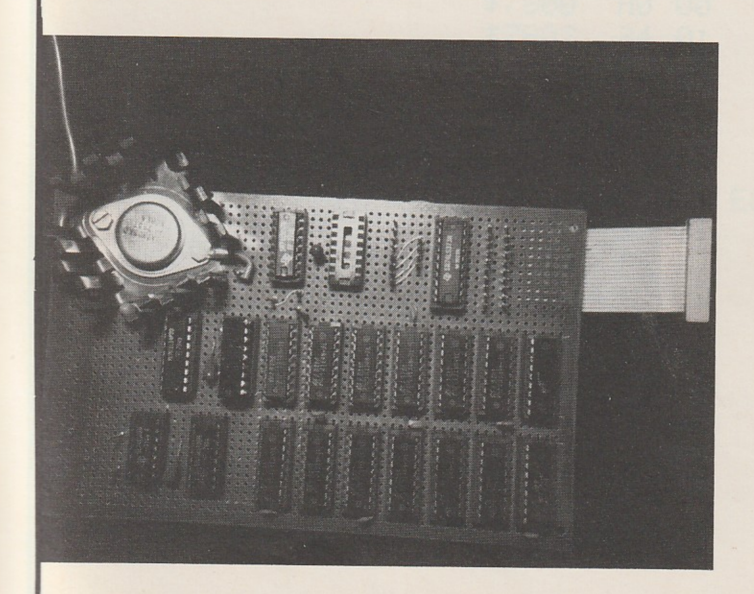
Les lignes d'activation du 74LS138 sont fixées actives donc br 4,5 à la masse, br 6 au + 5V. Les br 1,2 et 3 reçoivent les sorties (resp. br.4, 7 et 9) du 74LS157. Les entrées sont br. 2, 5 et 11

liées respectivement à A0, A1, A2 (br 8,7 et 6) de la 2316 et br 3, 6, 10 liées à PA0 - PA2 du port utilisateur. La ligne de sélection est liée à son homologue du 74LS-157, UF3 de la carte CBM (signal S), tandis que la ligne G* (br 15) est liée à PA3 (signal BR/HR*), comme l'est la ligne OE*, br 1 du 74LS373. La br 11 est liée à son homologue du 74LS373, UF9 de la carte CBM (signal appelé latch). Le branchement du 74LS00 est évident d'après l'article du numéro 5.

Le problème de l'alimentation

La carte HR contenant un nombre assez élevé de 2114, circuit qui consomme beaucoup, nous avons préféré la doter de son propre régulateur 5 V, avec son propre radiateur. Il travaille à partir du 9V non régulé présent sur la carte CBM : br 1 (celle qui est séparée des autres) du connecteur J11 (celui des connecteurs 7 broches, le long du radiateur, le plus proche du port IEEE).

La photo ci-dessous vous montre la carte effectivement réalisée, il n'est pas question de décrire dans La Commode des réalisations fictives et nos amis qui nous ont rendu visite au SICOB 1981 l'ont vu fonctionner effectivement. On reconnaît en haut à gauche le régulateur et son radiateur. Le fil qui part vers le haut va vers le connecteur J11. En haut, au milieu

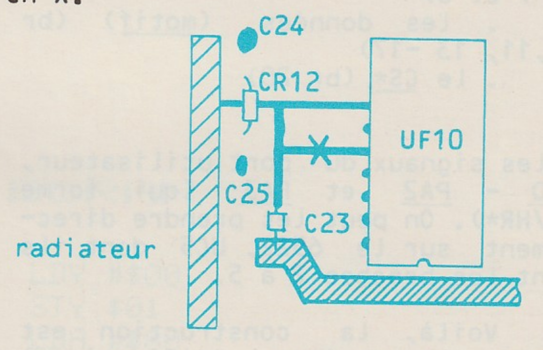


le support vide et les "jumpers" corespondent à des extensions non décrites ici. On voit à droite la nappe souple qui amène les 20 signaux pris sur la carte CBM : elle se connecte à un groupe de tiges à wrapper ajoutées à la carte CBM. Cette technique est préférable à la liaison directe car elle permet un démontage de la carte HR.

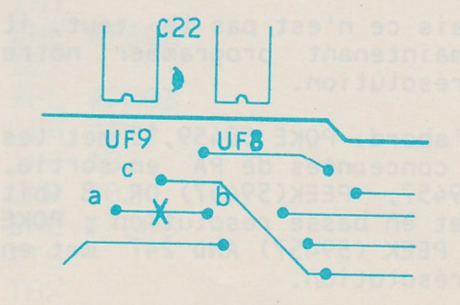
PREPARATION DE LA CARTE CBM - PRISE DES SIGNAUX

La première modification à effectuer sur la carte CBM est de monter les RAM écran (2114 UF7 et UF8) sur support si elles ne le sont pas. Pour cela, en l'absence de pompe à dessouder très puissante, le mieux est de sacrifier les deux 2114 en coupant les pattes à ras. Vous videz ensuite les trous à la tresse et installez les supports.

Il y a ensuite deux traces du circuit imprimé à couper. La première est celle qui amène le 0V à la broche 20 du générateur de caractères UF10. On a l'aspect suivant (côté composants) : couper en X.



La seconde, consiste à couper la ligne RVS pour former les signaux RVSA (amont) et RVSB (aval). On a, côté composants, un peu en avant entre UF9 et UF8 :



Couper en X. a est RVSA, b est RVSB. Notons dès maintenant que c est le signal latch (br 11 du 373).

Ceci étant, comme nous l'avons dit nous avons assuré la prise des signaux en collant sur la carte du CBM (à droite un peu en avant du connecteur J4 un petit morceau de carte pastillée et percée sur laquelle nous avons soudé 20 tiges à wrapper, au pas 2,54 en deux rangées de 10, ce qui permet d'y brancher un connecteur 3M pour nappe souple. La nappe souple est soudée à la carte HR. Il sufit d'amener aux 20 tiges les 20 signaux que nous avons à prélever sur la carte CBM (en plus de ceux qui sont prélevés sur les supports de 2114. Ces 20 signaux sont :

- latch, RVSA et RVSB qu'on a déjà vus

- S qu'on prend à la broche 1 du 74LS157, UF3

- signaux de la 2316 UF10 (générateur de caractères) :

. les 3 adresses A0 - A2 (br 8, 7 et 6)
. les données (motif) (br 10,11, 13 -17)
. le CS* (br 20)

- les signaux du port utilisateur, PA0 - PA2 et PA3 (qui forme BR/HR*). On peut les prendre directement sur le 6522, UC5 dont ils sont les broches 2 à 5.

Voilà, la construction est finie. La première vérification à faire de votre montage est qu'à la mise sous tension, il donne l'affichage basse résolution normal.

LOGICIEL

Mais ce n'est pas le tout, il faut maintenant programmer notre haute résolution.

D'abord, POKE 59459,15 met les lignes concernées de PA en sortie. POKE 59457, PEEK(59457) OR 8 (bit 3=1) met en basse résolution ; POKE 59457, PEEK (59457) AND 247 met en haute résolution.

Il suffit maintenant de savoir allumer ou éteindre le point des coordonnées élémentaires X,Y sur l'écran. Etant données X et Y, on calcule XM = INT(X)/8 : YM = INT(Y/8) : M = E + 40 * YM + XM ses coordonnées de maille. E vaut 32768.

On calcule ensuite la ligne élémentaire L et la colonne élémentaire correspondantes : L = X AND 7 : K = Y AND 7 (voir l'article de La Commode numéro 5 page 36).

Chacune des 8 lignes élémentaires possibles 0 à 7 correspond à un bloc mémoire de la carte HR. La case concernée dans le bloc a pour adresse M, mais il faut d'abord sélectionner le bloc voulu par POKE 59457,L.

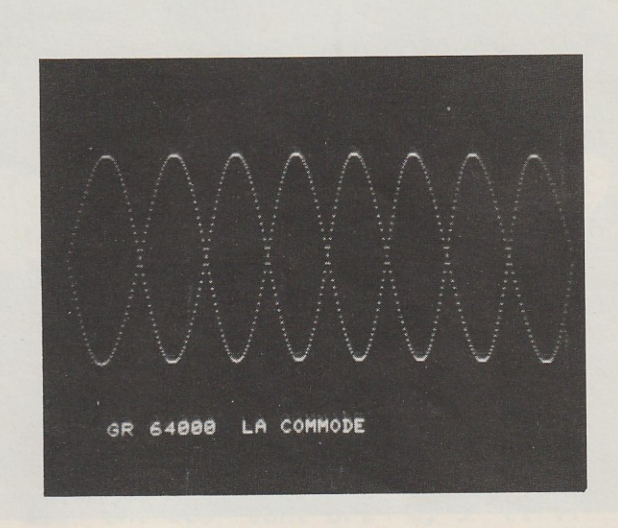
Ensuite

POKEM,PEEK(M)OR2↑(7-K)
allume le point

POKEM,PEEK(M)AND(255-2↑(7-K))
l'éteint.

Le listing qui suit est un exemple de tracé de deux sinusoides. En 20, on appelle une routine en langage machine qui met l'écran à blanc. De 100 à 140 on écrit un texte en bas de l'écran en supposant qu'on a en $7000, une copie du générateur de caractères. La photo montre l'affichage obtenu.

Daniel-Jean DAVID



HAUTE RESOLUTION

```
 5 DEFFNF(X)=75+60*SIN(X*π/40)
 7 DEFFNG(X)=75-60*SIN(X*π/40)
 10 POKE59459,15
 20 SYS30720
 40 X=-1
 50 FOR J=0TO39:FORKE=0TO7
 60 X=8*J+KE:Y=FNF(X):M=INT(Y/8):F=2↑(7-KE)
 65 Y2=FNG(X):M2=INT(Y2/8)
 70 R=INT(Y-8*M):POKE59457,R:E=32768+40*M+J
 75 K=PORPEEK(E):POKEE,K
 80 R2=INT(Y2-8*M2):POKE59457,R2:E2=32768+40*M2+J
 90 KK=PORPEEK(E2):POKEE2,KK:NEXT:NEXT
 100 FORLE=0TO7:POKE59457,LE
 110 PRINT"SQQQQQQQQQQQQQQQQQQQQQ]]]GR 64000  LA COMMODE":NEXT
 120 D=32767+40*21+4:FORI=DTOD+19
 130 FORLE=0TO7:POKE59457,LE:A=PEEK(I)
 140 B=PEEK(28672+8*A+LE):POKEI,B:NEXT:NEXT
 200 WAIT152,1:POKE59459,0
READY.
```

VIDAGE ECRAN

```
$7800  A0 00     LDY #$00
$7802  84 01     STY $01
$7804  A9 80     LDA #$80
$7806  85 02     STA $02
$7808  98        TYA
$7809  A2 07     LDX #$07
$780B  8E 41 E8  STX $E841
$780E  91 01     STA ($01),Y
$7810  C8        INY
$7811  D0 FB     BNE $780E
$7813  CA        DEX
$7814  10 F5     BPL $780B
$7816  A9 84     LDA #$84
$7818  E6 02     INC $02
$781A  C5 02     CMP $02
$781C  D0 EA     BNE $7808
$781E  60        RTS
$781F  AA        TAX
```

PETits trucs et COMBines

Des petites choses bien utiles...

<u>Complément à "manips magnéto"</u> (numéro 3, page 47)

Pour empêcher la routine d'interruption d'arrêter l'alimentation 6V des magnétos si on l'utilise, faire

```
PET POKE 519,1 (magnéto 1)
    POKE 520,1 (magnéto 2)

CBM POKE 249,1
    POKE 250,1

VIC POKE 192,1 (sous réserves)
```

Pierre Etienne THALBERG

<u>Quelques petits trucs sur cassettes</u>

Quand on écoute sur mini cassette normale une K7 enregistrée par le CBM, on entend l'enregistrement du programme.

Celui-ci se divise en plusieurs parties :

- sifflement régulier aigu: amorce (8 sec)

- son irrégulier s grave: le nom du fichier

- sifflement aigu (1/2 seconde)

- son irrégulier plus grave: le programme lui-même.

On peut utiliser ce fait pour positionner correctement la bande, au son, sur l'amorce par une K7 qui contient plusieurs programmes: gain de temps.

Si on n'a pas de lecteur de K7 audio, on peut utiliser une radio FM sur environ 104 MHZ (assez près du CBM). On entend, à la radio le même son grave que sur le lecteur de K7 (il convient de régler correctement la fréquence).

Ceci peut donner certaines indications sur ce qui se passe pendant un LOAD.

Exemple : Si on entend une annonce et un nom et rien ne s'affiche à l'écran c'est qu'il y a un problème (par exemple on a laissé PEEK(59467) = 16 = sortie son valide)

- Autre chose.

Pour savoir si une cassette est vierge : appuyer sur ❮ sans faire LOAD, mais en ayant enfoncé PLAY du magnéto.

Si il n'y a qu'un ❮ qui apparaît, la cassette est vierge. Si plusieurs ❮ apparaîssent, la K7 n'est pas vierge. Je n'ai pas trouvé pourquoi.

- Pour ne plus avoir de curseur POKE 144,47 (attention avec EDEX)

- PEEK (PEEK(151) + 59127) donne le

code ASCII de la touche enfoncée

- GET avec curseur : POKE 158,0 : POKE 167,0 : WAIT 158,1
  GET A$: POKE 167,1... suite

- Essayez WAIT 6502,100, c'est amusant.

Etienne LEGENDRE

<u>Programme pour changer le NOM et l'ID sur une disquette sans toucher à rien d'autre.</u>

J'utilise les commandes U1 et U2. Cela fonctionne parfaitement sur 4040. (et sans doute aussi sur 8050 pour la 2ème version, mais cette deuxième version n'a pas été testée, faute de matériel sous la main !)

Pierre THERY

```
10 PRINT"BASIC 4.0"
20 PRINT"CHANGER LE NOM ET L'ID D'UNE DISQUETTE"
30 PRINT"                              SUR 4040 "
50 INPUT"A QUEL DRIVE 0";DR$
60 INPUT"NOUVEAU NOM (16 CAR) ";N$
70 INPUT"IDENTIFICATEUR (2 CAR)  XX";I$:I$=LEFT$(I$,2)
80 PRINT""
90 INPUT" ETES-VOUS SUR (O/N) ";K$:IF K$=""THEN90
100 IF K$="N"THEN10
110 FOR I=1TO18-LEN(N$):N$=N$+CHR$(160):NEXT I
120 N$=N$+I$
130 OPEN80,8,15,"I"+DR$
140 OPEN82,8,2,"#"
150 PRINT#80,"U1:2,"+DR$+",18,0"
160 PRINT#80,"B-P"2;144
170 PRINT#82,N$;
180 PRINT#80,"U2:2,"+DR$+",18,0"
190 PRINT#80,"I"+DR$
200 PRINT"                         FIN "
READY.

READY.

10 PRINT"BASIC 4.0"
20 PRINT"CHANGER LE NOM ET L'ID D'UNE DISQUETTE"
30 PRINT"                              SUR 8050 "
50 INPUT"A QUEL DRIVE 0";DR$
60 INPUT"NOUVEAU NOM (16 CAR) ";N$
70 INPUT"IDENTIFICATEUR (2 CAR)  XX";I$:I$=LEFT$(I$,2)
80 PRINT""
90 INPUT" ETES-VOUS SUR (O/N) ";K$:IF K$=""THEN90
100 IF K$="N"THEN10
110 FOR I=1TO18-LEN(N$):N$=N$+CHR$(160):NEXT I
120 N$=N$+I$
130 OPEN80,8,15,"I"+DR$
140 OPEN82,8,2,"#"
150 PRINT#80,"U1:2,"+DR$+",39,0"
160 PRINT#80,"B-P"2;6
170 PRINT#82,N$;
180 PRINT#80,"U2:2,"+DR$+",39,0"
190 PRINT#80,"I"+DR$
200 PRINT"                         FIN "
READY.
```

Super Yam

Une variante du jeu de YAM'S

Règle du jeu :

Le jeu se joue avec 5 dés et 3 lancers. Il s'agit, comme dans le jeu de Yam's, de réaliser un certain nombre de combinaisons :

- de 1 à 6 : maximum de dés du numéro, valeur : somme des dés du numéro choisi.
-Full: 3 dés semblables + 2 dés semblables ; 30 pts.
-Carré: 4 dés semblables;30 pts.
-Quinte: 5 dés consécutifs;40 pts.
-Yam: 5 dés semblables;50 pts.
-Maxi: total des dés le plus élevé possible ; somme des points.
-Mini: idem, mais inférieur au maxi.

Avoir 61 ou plus au total de 1 à 6 donne un bonus de 30 points.

L'objectif est de faire le maximum.

Mais, à la différence du Yam's, le Super-Yam se joue sur trois tableaux :
- Montée : où on ne peut enregistrer les combinaisons que de bas en haut.
- Descente : où on ne peut enregistrer les combinaisons que de haut en bas.
- Libre : où on choisit n'importe quelle combinaison.

Fonctionnement du programme :

- Début : introduire :
  - nombre de joueurs
  - nom des joueurs.

- Partie : Lorsque les dés sont arrêtés, l'écran demande "Voulez-vous voir la marque ?" : réponse par O/N (GET). Lorsque vous avez vu la marque, le retour aux dés est obtenu en tapant une touche quelconque (GET). L'écran affiche alors "Combien de dés rejouez-vous?" : réponse 0 à 5 (GET ; si la réponse est de 1 à 4, l'écran affiche "quels dés ?" : votre réponse est le numéro d'ordre des dés que vous voulez rejouer, sachant qu'ils sont implicitement numérotés de 1 à 5, en allant de gauche à droite sur l'écran.

Après le 3ème lancer, ou avant si vous avez rejoué 0 dé, l'écran demande "sur quel tableau jouez-vous ?" : réponse M (montée), D(descente) ou L (libre) (GET). Pour M ou D, le jeu sera enregistré à la position correcte, mais si vous répondez L, il vous sera demandé "Quelle combinaison choisissez-vous ?"; vous devez répondre un nombre de 1 à 6, FULL, CARRE, QUINTE, YAM, MAXI ou MINI, puis Return.

- Fin de partie : le tableau final apparaît, suivi du score final et du % par rapport au score idéal. Appuyer sur une touche fera apparaître "Voulez-vous jouer une autre partie ?" : réponse O/N (GET).

Claude COSSU

```
10 PRINT"JEU DE YAM"
20 PRINT"=========="
30 INPUT"NOMBRE DE JOUEURS";M
40 FORI=1TOM:PRINT"NOM DU JOUEUR"I;:INPUTM$(I)
50 IFLEN(M$(I))<6THENM$(I)=M$(I)+" ":GOTO50
60 NEXT
140 N=3:A$(1)="MONTEE":A$(2)="DESCENTE":A$(3)="LIBRE"
200 DIMZ$(12),C%(M,3,12),T%(M,3,12),JJ(36),G(12),X$(6),J(12),C$(3,12)
202 DIMCS(3),CX(3),S$(3),CZ(M,3),CT(3),W$(3),U$(3),LL(M),KK(M)
210 FORJ=1TO12:READZ$(J):NEXT
220 DATA"1","2","3"," 4","5","6"
222 DATAFULL  ,CARRE ,QUINTE,YAM   ,MAXI  ,MINI  ,
240 X$(1)="    ":X$(2)="     "
245 X$(3)="     ":X$(6)="     "
250 X$(4)="     ":X$(5)="     "
255 FORA=1TOM:LL(A)=12:KK(A)=1:NEXT
260 FORA=1TOM:FORJ=1TO12:FORI=1TON:C%(M,I,J)=0:T%(M,I,J)=0:NEXT:NEXT:NEXT
261 FORJJ=1TO36
262 FORA=1TOM
270 KM=5:L=5
275 FORH=1TO5:F(H)=H:NEXTH
280 FORK=1TO3:PRINT"JOUEUR "M$(A):PRINT"COUP NUMERO"JJSPC(10);
285 PRINT"LANCER NUMERO"K""
287 FORW=1TO20:IFW>1THENPRINT""
288 T%=PEEK(143)
290 FORH=1TOL:B(F(H))=INT(6*RND(T%+H)+1):NEXTH
300 FORD=1TOKM:PRINT""X$(B(D))"";:NEXTD:PRINT"":NEXTW
305 GOSUB2900
310 IFK=3GOTO350
320 PRINT"COMBIEN DE DES REJOUEZ-VOUS? "
321 GETL$:IFL$<>"0"ANDL$<>"1"ANDL$<>"2"ANDL$<>"3"ANDL$<>"4"ANDL$<>"5"THEN321
322 IFL$="0"THEN350
323 IFL$="5"GOTO340
330 L=VAL(L$):PRINT"QUELS DES ?":FORE=1TOL
331 GETF$
332 IFF$<>"1"ANDF$<>"2"ANDF$<>"3"ANDF$<>"4"ANDF$<>"5"THEN331
334 F(E)=VAL(F$)
335 NEXTE
340 NEXTK
350 PRINT"SUR QUEL TABLEAU JOUEZ-VOUS(M,D,L)"
351 GETWZ$
352 IFWZ$<>"L"ANDWZ$<>"M"ANDWZ$<>"D"THEN351
```

```
355 IFWZ$="L"THEN400
360 IFWZ$="M"GOTO364
362 IFWZ$="D"GOTO369
363 PRINT"ERREUR":GOTO350
364 II=1:IFLL(A)<1GOTO363
365 Y$=Z$(LL(A)):IFLL(A)<7THENY$=MID$(Y$,4,1)
366 LL(A)=LL(A)-1:GOTO410
369 II=2:IFKK(A)>12GOTO363
370 Y$=Z$(KK(A)):IFKK(A)<7THENY$=MID$(Y$,4,1)
372 KK(A)=KK(A)+1:GOTO410
400 II=3:PRINT"QUELLE COMBINAISON CHOISISSEZ-VOUS";:INPUTY$
410 FORG=1TO6:IFY$=MID$(Z$(G),4,1)GOTO440
415 NEXT
420 FORG=7TO12:IFLEFT$(Y$,3)=LEFT$(Z$(G),3)THEN440
425 NEXT
430 PRINT"COMBINAISON INCONNUE":GOTO400
440 IFT%(A,II,G)=1THENPRINT"COMBINAISON DEJA CHOISIE":GOTO350
450 I=II:J=G:T%(A,I,J)=1:FORW=1TO5:R(W)=B(W):NEXT
460 FORV=1TO5
470 JM=1:MM=R(JM)
480 FORW=2TO5:IFR(W)<=MMTHEN500
490 JM=W:MM=R(W)
500 NEXTW
510 PO(V)=JM:R(JM)=-1
520 NEXTV
540 ONJGOSUB3300,3300,3300,3300,3300,3300,3410,3500,3610,3710,3810,3910
580 NEXTA
585 NEXTJJ
599 FORA=1TOM:GOSUB3000
600 VV=CZ(A,1)+CZ(A,2)+CZ(A,3):SM=1032:MX=INT(100*VV/SM+.5)
610 PRINT"SCORE:"VV"MAXI.:"SM"TAUX:"STR$(MX)+"%":GOSUB5000
615 NEXT
618 PRINT"VOULEZ-VOUS JOUER UNE AUTRE PARTIE ?"
619 GETA$:IFA$<>"N"ANDA$<>"O"THEN619
620 IFA$="O"GOTO255
630 END
2800 PRINT"JOUEUR "M$(A):PRINT"COUP NUMERO"JJSPC(10);
2810 PRINT"LANCER NUMERO"K""
2820 FORD=1TOKM:PRINT""X$(B(D))"TT";:NEXTD:PRINT"":RETURN
2900 PRINT"VOULEZ-VOUS VOIR LA MARQUE ?(O/N)"
2902 GETQ$
2905 IFQ$<>"O"ANDQ$<>"N"THEN2902
```

```
2910 IFQ$="N"THENRETURN
3000 PRINT"┌──────┬──────┬──────┬──────┐"
3005 MM$(A)="▚"+LEFT$(M$(A),6)+"■"
3010 PRINT "|"MM$(A)"| MONTEE |DESCENTE| LIBRE |"
3020 PRINT "└──────┴──────┴──────┴──────┘"
3032 FORG=1TO6
3040 PRINT"|"Z$(G)"|";:FORI=1TON:OQ=C%(A,I,G):GOSUB6000:NEXTI:PRINT""
3045 NEXTG
3046 FORH=1TO32:PRINT"─";:NEXT:PRINT""
3050 FORI=1TON:CS(I)=0:FORG=1TO6:CS(I)=CS(I)+C%(A,I,G):NEXTG
3056 NEXTI
3057 G=13
3060 PRINT"| TOTAL1 |";:FORI=1TON:OQ=CS(I):GOSUB6000:NEXTI:PRINT""
3065 FORI=1TON:T(I)=0:IFCS(I)>60THENT(I)=30
3068 NEXTI
3070 PRINT"| BONUS  |";:FORI=1TON:OQ=T(I):GOSUB6000:NEXT:PRINT""
3075 FORI=1TON:CT(I)=CS(I)+T(I)
3078 NEXTI
3080 PRINT"| TOTAL2 |";:FORI=1TON:OQ=CT(I):GOSUB6000:NEXTI:PRINT""
3090 FORH=1TO32:PRINT"─";:NEXT:PRINT""
3100 FORG=7TO12
3120 PRINT"|"Z$(G)"|";:FORI=1TON:OQ=C%(A,I,G):GOSUB6000:NEXTI:PRINT""
3130 NEXTG
3140 FORH=1TO32:PRINT"─";:NEXT:PRINT""
3150 FORI=1TON
3155 CX(I)=0
3160 FORG=7TO12:CX(I)=CX(I)+C%(A,I,G):NEXTG
3170 CZ(A,I)=CT(I)+CX(I)
3178 NEXTI
3180 PRINT"| TOTAL3 |";:FORI=1TON:OQ=CZ(A,I):GOSUB6000:NEXTI:PRINT""
3190 PRINT"└──────┴──────┴──────┴──────┘":IFJJ<36THENGOSUB5000
3195 IFJJ<37THENGOSUB2800
3200 RETURN
3300 FORV=1TO5:IFB(PO(V))=JTHENC%(A,I,J)=C%(A,I,J)+B(PO(V))
3315 NEXTV
3320 RETURN
3410 IFB(PO(1))=B(PO(2))ANDB(PO(2))=B(PO(3))ANDB(PO(4))=B(PO(5))THENC%(A,I,J)=3
0
3420 IFB(PO(1))=B(PO(2))ANDB(PO(3))=B(PO(4))ANDB(PO(4))=B(PO(5))THENC%(A,I,J)=3
0
3430 RETURN
3500 FORV=1TO5
```

```
 3510 IFB(PO(1))=B(PO(2))ANDB(PO(2))=B(PO(3))ANDB(PO(3))=B(PO(4))THENC%(A,I,J)=3
0
 3520 IFB(PO(2))=B(PO(3))ANDB(PO(3))=B(PO(4))ANDB(PO(4))=B(PO(5))THENC%(A,I,J)=3
0
 3525 NEXTV
 3530 RETURN
 3610 FORV=1TO4:IFB(PO(V+1))=B(PO(V))-1THENNEXT
 3615 IFV=5THENC%(A,I,J)=40
 3620 RETURN
 3710 FORV=1TO4:IFB(PO(V+1))=B(PO(V))THENNEXTV:IFV=5THENC%(A,I,J)=50
 3720 RETURN
 3810 FORV=1TO5:C%(A,I,J)=C%(A,I,J)+B(PO(V)):NEXT
 3820 IFT%(A,I,12)=1ANDC%(A,I,12)>=C%(A,I,J)THENC%(A,I,J)=0
 3830 RETURN
 3910 FORV=1TO5:C%(A,I,J)=C%(A,I,J)+B(PO(V)):NEXT
 3920 IFT%(A,I,11)=1ANDC%(A,I,11)<=C%(A,I,J)THENC%(A,I,J)=0
 3930 RETURN
 5000 GETA$:IFA$=""THEN5000
 5010 RETURN
 6000 IFG=13ANDOQ=0THENPRINT"         |";:GOTO6020
 6002 IFG=13GOTO6010
 6005 IFT%(A,I,G)=0THENPRINT"         |";:GOTO6020
 6010 PRINTSPC(5-LEN(STR$(OQ)))OQSPC(1)" |";
 6020 RETURN
READY.
```

Bourse d'échange de programmes

"Graphique facile" pour VIC 20 + extension 3 K + Joystick ATARI.

Le programme permet la définition aisée des caractères graphiques en disposant des commandes suivantes :

- Rapidité de la construction du caractère et des corrections grâce au joystick.

- Visualisation des caractères du VIC par appel de son code

- Mise en contraste inversé

- obtention du symétrique

- essai du caractère en grandeur réelle

- Liste des nombres de définition à mettre en DATA.

"RUBIK'S CUBE" pour VIC 20 + extension 16 K

Ecrit en BASIC, de taille 13 K octets, prévu pour être utilisé avec un poste noir et blanc ou couleur.

Après avoir introduit les facettes du cube, vous pouvez le manipuler à l'aide du clavier ou bien le faire reconstituer par l'ordinateur suivant l'un des deux modes :

- lent avec affichage des faces

- rapide avec un codage simple des rotations qui permet sans entraînement une cadence de 1 coup par seconde (cadence réglable).

Hervé LE MARCHAND
Ecole CENTRALE
92250 CHATENAY MALABRY

Je suis l'auteur du programme "gestion de notes" dont vous avez publié les caractéristiques dans la rubrique "bourse d'échange" du numéro 5.

Depuis, je me suis équipé d'une unité de disquette CBM 2031 et de la mémoire Edex version 3.3.

J'ai repris ce programme pour l'adapter à ce nouveau matériel. Il est ainsi maintenant beaucoup plus performant.

Je vous joins un court document de ce que l'on peut sortir avec ce programme (dont je suis très fier !!!) c'est mon premier gros programme.

Pour terminer, je vous signale que je viens de terminer un programme sur les séries de Fourrier, qui permet de visualiser la reconstruction d'un signal périodique (créneau, dent de scie, etc.) à partir du fondamental et des harmoniques successifs. Je le

tiens à disposition pour la bourse d'échanges.

J.C. RAIMBAULT
14 avenue Maréchal FOCH
63120 COURPIERE

Moyennes et rang des élèves par ordre alphabétique

moyennes steno

nom	moy.	rg	*	nom	moy.	rg
Bara	06.00	18	*	Mye	abs	
Destritea	13.50	3	*	Pianaro	06.50	16
Dubourgno	09.50	11	*	Rigaud	11.50	8
Fedit	15.00	2	*	Ronzier	12.50	6
Gourbeyre	09.00	13	*	Roux	10.50	9
Laclemenc	13.50	3	*	Pommerett	abs	
Lagrange	10.50	9	*	Thery	13.00	5
Laidaoui	08.00	15	*	Thonier	15.50	1
Monier	06.50	16	*	Venel	09.50	11
Monat	08.50	14	*	Voulhoux	12.50	6

pour continuer taper return

ici par rangs croissants

moyennes steno

nom	moy.	rg	*	nom	moy.	rg
Thonier	15.50	1	*	Dubourgno	09.50	11
Fedit	15.00	2	*	Venel	09.50	11
Destritea	13.50	3	*	Gourbeyre	09.00	13
Laclemenc	13.50	3	*	Monat	08.50	14
Thery	13.00	5	*	Laidaoui	08.00	15
Ronzier	12.50	6	*	Monier	06.50	16
Voulhoux	12.50	6	*	Pianaro	06.50	16
Rigaud	11.50	8	*	Bara	06.00	18
Lagrange	10.50	9	*	Mye	abs	
Roux	10.50	9	*	Pommerett	abs	

pour continuer taper return

Notes d'un devoir, ici le premier devoir de sténo

steno 1

Bara	06.0	*	Mye	abs
Destriteaux	13.5	*	Pianaro	06.5
Dubourgnoux	09.5	*	Rigaud	11.5
Fedit	15.0	*	Ronzier	12.5
Gourbeyre	09.0	*	Roux	10.5
Laclemence	13.5	*	Pommerette	abs
Lagrange	10.5	*	Thery	13.0
Laidaoui	08.0	*	Thonier	15.5
Monier	06.5	*	Venel	09.5
Monat	08.5	*	Voulhoux	12.5

voulez-vous consulter un autre fichier ?
(o ou n)

Pet Club

COMMUNIQUES DE PRESSE

Face au développement très rapide de leur activité, les fabricants, constructeurs et distributeurs de microordinateurs n'ont bien souvent pas la possibilité de traiter leur documentation dans les meilleures conditions.

EDIMICRO leur propose un service complet dans ce domaine :

- TRADUCTION des documents par des spécialistes de la microinformatique, qui peuvent remettre les textes sur disquettes de traitement de texte.

- REWRITING et mise en forme par des experts du sujet traité, qui sont aussi des pédagogues.

- EDITION et IMPRESSION sous la forme appropriée.

Pour tous renseignements :

F.D.S. Département EDIMICRO
10, rue Henri Pape
75013 PARIS

Tél. : 588 76 53

FDS

ENFIN, UN CYCLE COMPLET DE SEMINAIRES MICROORDINATEURS POUR LES UTILISATEURS NON SPECIALISTES !

F.D.S. Formation a le plaisir de vous présenter son programme exceptionnel de séminaires, qui seront donnés pour la première fois en Octobre/Novembre 1982.

- CHOIX D'UN MICROORDINATEUR POUR L'ENTREPRISE

- TABLEAUX FINANCIERS (VISICALC)

- INTRODUCTION AU TRAITEMENT DE TEXTE

- INTRODUCTION AUX BASE DE DONNEES

- LA TELEMATIQUE : Techniques et utilisations

- LE BASIC GRAPHIQUE

- PREMIERS PAS EN BASIC

- PERFECTIONNEMENT ET T.P. EN BASIC

- LE GENIE LOGICIEL

- LE SYSTEME D'EXPLOITATION MS-DOS

- LE SYSTEME D'EXPLOITATION UNIX-XENIX

F.D.S. : 588.76.53
10, rue Henri Pape
75013 PARIS

CLUB EPS

Les réunions hebdomadaires du Club EPS ont repris à partir du lundi 27 septembre 1982 à 18 h 30, sous la responsabilité de Monsieur Pierre LESTIENNE.

La cotisation est de 500 francs par an ou de 200 francs par trimestre.

Pour tout renseignement, téléphoner à Pierre LESTIENNE.

(1) 523 32 20

ou écrire à

Club Micro EPS
45 rue des Petites Ecuries
75010 PARIS.

INTERESSES PAR LA FCCF ET UN CLUB

Monsieur VIDAUD Jean René
6, résidence La Source
78870 BAILLY

Monsieur DAVITTE Alain
102, rue Maurice Arnoux
92120 MONTROUGE

Monsieur Hervé LE MARCHAND
Résidence ECP 6015
Avenue Sully Prudhomme
92250 CHATENAY MALABRY

Monsieur DISSON
6, rue Jean Raynal
91390 MORSANG SUR ORGE

Complétez votre Collection

Les numéros anciens de La Commode sont encore disponibles mais peut-être pas pour longtemps. Hâtez vous de compléter votre collection si vous ne voulez pas rater des renseignements essentiels.

BON DE COMMANDE

à envoyer à La Commode 28, rue Vicq-d'Azir 75010 PARIS

Je désire les numéros suivants de La Commode

☐ numéro 1 ☐ numéro 2 ☐ numéro 3 ☐ numéro 4 ☐ numéro 5

Joignez votre réglement (35F par numéro) à l'ordre de SEDERMI

☐ CCP ☐ CB ☐ MANDAT

Bientôt épuisés

NOM ....
ADRESSE ....
....
....

LE FEEDBACK DE LA COMMODE

Pour produire une revue qui vous donne toujours plus de satisfaction, nous avons absolument besoin de vos réponses à ces questionnaires.

Il y a un questionnaire pour chaque numéro. Parmi les réponses concernant chaque numéro, deux seront tirées au sort et leurs auteurs recevront un petit souvenir de La Commode.

Comme au patinage artistique, vous attribuez deux notes à chaque article: une note sur l'intérêt du sujet et une note sur la façon dont le sujet est traité (rédaction, clarté, etc...).Voici les barêmes:

sujet :		traitement :	
intérêt capital	4	excellent	4
très intéressant	3	bon	3
assez intéressant	2	moyen	2
pas d'intérêt pour moi	1	mauvais	1
n'a pas, d'après moi, sa place dans La Commode	0	nul	0

Numéro 6 titre	note sujet	note traitement
Courrier des lecteurs		
Magazine		
Premier contact avec le Commodore 64		
Banc d'essai de deux produits Chatain-Blanchon		
Qui a dit que le BASIC n'était pas structuré ?		
Périphériques de jeux sur le VIC		
Correspondances VIC/CBM et CBM/VIC		
Banc d'essai de la cartouche VIC PROGRAMMER's AID		
Gestion de comptes		
VIC à brac		
Affichage haute résolution		
Petits trucs et combines		
Jeu de SUPER YAM		
Bourse d'échange des programmes		
PET CLUB		
Bibliographie		
Petites annonces		

Autres commentaires : ....
....
....

Petit questionnaire (réponses indispensables pour être tiré au sort).

nom : .... âge : .....
adresse : ....
....
profession : ....

J'utilise un Commodore modèle ....
pour:
ma profession mes loisirs

Précisez SVP, si vous utilisez différents modèles.

A renvoyer à : La Commode, service FB
28, rue Vicq-d'Azir
75010 PARIS

Bibliographie

A propos de VIC REVEALED de Nick Hampshire. Disons tout de suite que la "révélation" ne sera pas instantanée et qu'il vous faudra beaucoup de relectures et de recherches pour utiliser efficacement les informations réparties sur 267 pages. Certes, tout y est, mais l'auteur ne s'embarrasse d'aucune pédagogie pour approfondir des notions qu'il suppose déjà connues du lecteur. On y trouve quelques rappels sur le langage machine, le fonctionnement du BASIC, les adresses des routines-système, la haute résolution simple et multicouleur, l'étude du 6522, les entrées-sorties, le magnétophone, le clavier, les RS 232 et IEEE, joysticks et paddles. La densité des informations est telle que les exemples sont pratiquement inexistants. En appendices sont donnés divers tables et schémas. On y remarquera la présentation judicieuse des codes CBM ainsi que la description d'une routine permettant d'enrichir l'interpréteur de nouvelles commandes. Il faut cependant l'adapter car elle est prévue pour une extension ROM.

Prix : 227 francs chez Ellix

Hervé LE MARCHAND

Notre confrère l'Ordinateur Individuel vient de publier un numéro spécial intitulé "l'Ordinateur de Jeux". On y trouvera en particulier un banc d'essai du VIC du point de vue de ses applications jeux.

Petites annonces

VENDS PET 2001 anciennes ROMS avec extension TOOLKIT + Doc + importante cassettothèque.

Commodore club Méridional La Fauvière Bât. A1
13010 MARSEILLE Tél. 35-50-72

Le Commodore Club Méridional recherche tout matériel électronique, petite mécanique, documentation pour réalisation au sein du club.

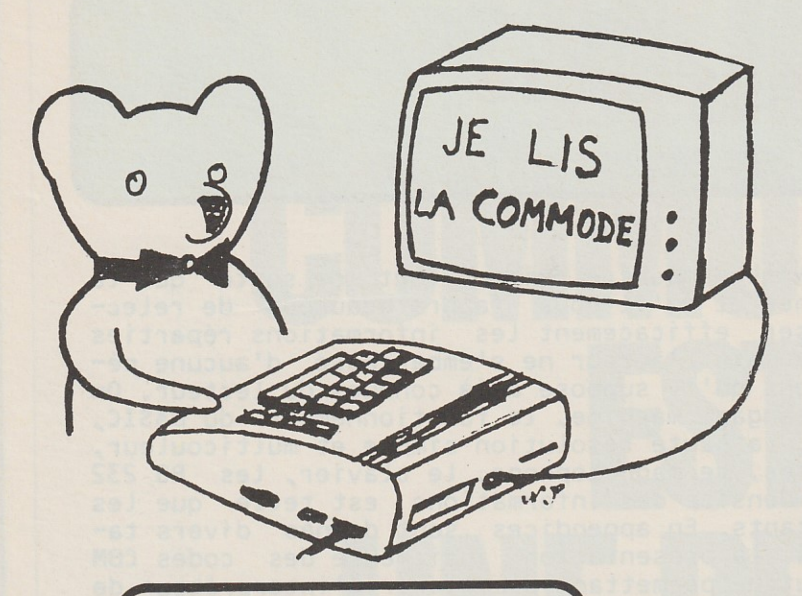
Industriels, distributeurs, offrez nous les matériels déclassés qui vous encombrent.

Commodore Club Méridional
La Fauvière Bât A1
13010 MARSEILLE Tél.35-50-72

Le Commodore Club Méridional a repris ses cours de basic pour débutant, venez nous rejoindre les 2ème Vendredi et dernier Vendredi de chaque mois au lycée Lacordaire. Boulevard Lacordaire à Saint-Just (Marseille) à partir de 20 h 30.

La Commode

2ᵉ année



Diffusion :
Ed. du P.S.I.
41-51, rue Jacquard
BP 86
77400 LAGNY

Publicité :
FORCE 7
41, rue de la Grange aux Belles
75483 PARIS CEDEX 10
Tél. 238.66.10

Publié par SEDERMI SARL
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

Rédaction

Rédacteur en chef :
Daniel TRECOURT

Rédacteur en chef adjoint :
Marc VAN CRAEYNEST

Chefs de rubriques :
Emile AMZALLAG
Jacques COUTHURES
Jean DELAVILLE
Pierre-Etienne THALBERG
François VULQUIN

Secrétaire de rédaction :
Chantal DESCHAMPS

Editeur :
Jean-Luc DESCHAMPS

Directeur de la publication :
Daniel-Jean DAVID

Rédaction-vente-abonnements :
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

Bulletin d'abonnement

A envoyer à La Commode
28, rue Vicq d'Azir - 75010 PARIS

Je désire m'abonner à la Commode (1 an, 4 numéros) à partir du numéro ....

NOM : .... PRÉNOM : ....

Adresse : ....

....

Signature

Règlement 120 F à l'ordre de SEDERMI :

☐ CCP ☐ CB ☐ ESPECES

Réabonnement ☐
Indiquez si possible votre ancien numéro d'abonné

Imprimerie Laballery, 58500 Clamecy. Imprimé en France. Directeur de la publication Daniel-Jean David
Dépôt légal : janvier 1983 - Numéro de commission paritaire : en cours. ISSN 0292-5079

INDISPENSABLE :

CLEFS POUR LE PET/CBM

Clefs pour le PET/CBM
par Daniel-Jean David

C'est l'aide mémoire de tout programmeur sur PET/CBM, il renferme toutes les informations de référence à retrouver rapidement : syntaxe des commandes, codes caractères, messages d'erreurs, codes machine, brochages, bonnes adresses. Il se termine par un recueil de 40 "trucs" utiles, les "Comment?.."

112 pages - 82,00 FF/680,00 FB

AGRAPH

Envoyer ce bon accompagné de votre règlement à **EDITIONS DU P.S.I.**

M.L.C.12

Je désire recevoir ________ exemplaires de CLEFS POUR LE PET/CBM au prix unitaire de 82,00 FF (630,00 FB) taxes, port et emballage compris (par avion ajouter 8 FF (75 FB) par livre).

TOTAL	

NOM ________________ PRENOM ________________

rue ________________ N° ________

Code post. |_|_|_|_|_| Ville ________________



Editions du P.S.I.
41-51, rue Jacquard
BP 86 - 77400 Lagny-s/Marne
Téléphone (6) 007.59.31